

## Patent Abstracts of Japan

PUBLICATION NUMBER : 09034755  
PUBLICATION DATE : 07-02-97

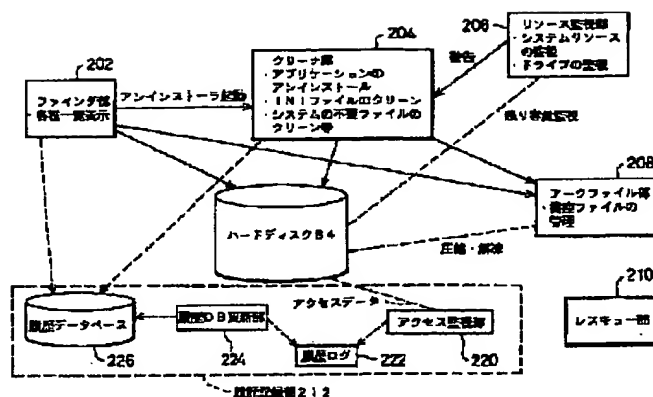
APPLICATION DATE : 15-03-96  
APPLICATION NUMBER : 08087703

APPLICANT : EE I SOFT KK;

INVENTOR : TOGASHI HIRONORI;

INT.CL. : G06F 12/00 G06F 9/06 G11B 20/00

TITLE : SYSTEM AND METHOD FOR  
ARRANGING STORAGE DEVICE  
CONTENTS



ABSTRACT : PROBLEM TO BE SOLVED: To increase the empty capacitance of an external storage device by facilitating the selection of a file to be deleted.

SOLUTION: A finder part 202 has a function for displaying the list of files stored in a hard disk 84 in various modes. A user grasps the use conditions of the hard disk 84 while utilizing this list display and can retrieve and delete unwanted files. In the case of deleting any application program or unwanted file, a cleaner part 204 has a function for performing the deletion of the application program, the deletion of the unwanted file and the deletion of information in the prescribed file or the like. When deleting any execution file, the cleaner part 204 analyzes various files to be used for the execution of the execution file and has a function for displaying those files as deleting candidates. A history recording part 212 records the access history of each file and supplies the access history of each file corresponding to the request from the finder part 202 or the cleaner part 204.

COPYRIGHT: (C)1997,JPO

(19) 日本国特許庁 (J P)

## (12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-34755

(43) 公開日 平成9年(1997)2月7日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 12/00	5 0 1	7623-5B	G 0 6 F 12/00	5 0 1 P
		7623-5B		5 0 1 E
9/06	4 1 0		9/06	4 1 0 B
G 1 1 B 20/00			G 1 1 B 20/00	Z

審査請求 未請求 請求項の数42 F D (全 37 頁)

(21) 出願番号 特願平8-87703

(22) 出願日 平成8年(1996)3月15日

(31) 優先権主張番号 特願平7-145397

(32) 優先日 平7(1995)5月18日

(33) 優先権主張国 日本 (J P)

(71) 出願人 594067221

エー・アイ ソフト株式会社

長野県諏訪市大和三丁目3番5号

(72) 発明者 佐藤 昭弘

長野県松本市中央二丁目1番27号 エー・

アイ ソフト株式会社内

(72) 発明者 武井 なほ子

長野県松本市中央二丁目1番27号 エー・

アイ ソフト株式会社内

(72) 発明者 水田 希世志

長野県松本市中央二丁目1番27号 エー・

アイ ソフト株式会社内

(74) 代理人 弁理士 五十嵐 孝雄 (外3名)

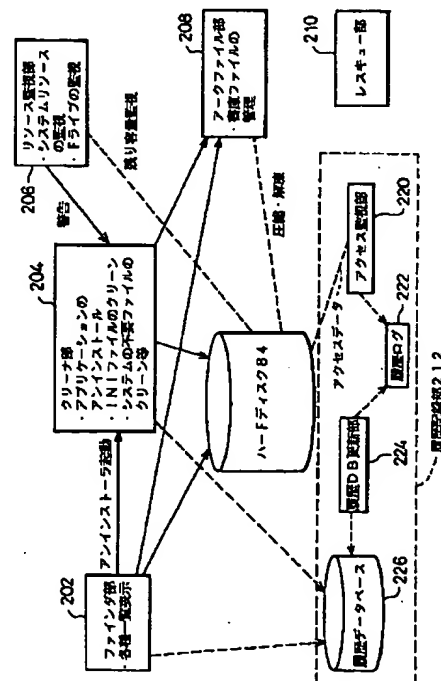
最終頁に続く

(54) 【発明の名称】 記憶装置内容整理システムおよび記憶装置内容整理方法

(57) 【要約】

【課題】 削除すべきファイルの選択を容易にして、外部記憶装置の空き容量を増加させる。

【解決手段】 ファインダ部202は、ハードディスク84内に格納されているファイルを各種のモードで一覧表示する機能を有する。ユーザは、これらの一覧表示を利用してハードディスク84の使用状況を把握し、不要なファイルを検索して削除することができる。クリーナ部204は、アプリケーションプログラムの削除や、不要ファイルの削除、所定のファイル内の情報の削除等を行なう機能を有している。クリーナ部204は、実行ファイルの削除を行なう場合に、実行ファイルの実行に伴って使用される種々のファイルを解析して、削除候補として表示する機能も有する。履歴記録部212は、各ファイルのアクセス履歴を記録して、ファインダ部202やクリーナ部204からの要求に応じて各ファイルのアクセス履歴を供給する。



## 【特許請求の範囲】

【請求項1】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納されているファイルを分類するための分類情報を入力するための分類情報入力手段と、前記分類情報に応じて、前記外部記憶装置に格納されているファイルを複数のファイル群に分類するとともに、各ファイル群のデータ総量を表示装置に表示する分類手段と、表示された前記複数のファイル群の1つを選択するためのファイル群選択手段と、前記選択手段によって選択されたファイル群に属するファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする記憶装置内容整理システム。

【請求項2】 請求項1記載の記憶装置内容整理システムであって、前記分類情報は、ファイルの拡張子別にファイルを分類する旨の情報を含む、記憶装置内容整理システム。

【請求項3】 請求項1記載の記憶装置内容整理システムであって、前記分類情報は、ディレクトリ単位でファイルを分類する旨の情報を含み、前記削除実行手段は、前記選択手段で選択されたディレクトリおよび該ディレクトリに含まれるファイルを削除する手段を備える、記憶装置内容整理システム。

【請求項4】 請求項1ないし3のいずれかに記載の記憶装置内容整理システムであって、さらに、前記外部記憶装置内の各ファイルへのアクセス回数とアクセス日時を含むアクセス履歴を記録する履歴記録手段を含み、前記分類情報は、前記履歴記録手段によって記録されたアクセス履歴に関する分類条件を示す情報を含む、記憶装置内容整理システム。

【請求項5】 請求項1ないし4のいずれかに記載の記憶装置内容整理システムであって、さらに、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合には、前記実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示する検索手段を備える、記憶装置内容整理システム。

【請求項6】 請求項5記載の記憶装置内容整理システムであって、前記検索手段は、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合に、前記実行ファイルに関して所定のファイル内に含

まれている情報を削除候補とする手段を備える、記憶装置内容整理システム。

【請求項7】 請求項6記載の記憶装置内容整理システムであって、前記削除候補となる情報は、WIN.INI ファイルおよび/またはレジストリファイル内の関連付け情報と、WIN.INI ファイル内のアプリケーションセクションと、前記レジストリファイル内のOLE 情報と、アイコンの登録情報とを含む、記憶装置内容整理システム。

【請求項8】 請求項6または7記載の記憶装置内容整理システムであって、さらに、前記削除候補となる情報を含むクリーン情報ファイルを作成する手段を備える、記憶装置内容整理システム。

【請求項9】 請求項5ないし8のいずれかに記載の記憶装置内容整理システムであって、前記削除候補のファイルの1つを内容表示対象ファイルとして選択する表示対象ファイル選択手段と、前記内容表示対象ファイルがアプリケーションプログラムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示するファイル内容表示手段、を備える記憶装置内容整理システム。

【請求項10】 請求項5ないし9のいずれかに記載の記憶装置内容整理システムであって、さらに、前記検索手段によって検索された削除候補の中から、削除対象を選択する削除対象選択手段と、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによっては復元されないファイルを簡易バックアップ対象ファイルとして特定する簡易バックアップファイル特定手段と、前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップするバックアップ実行手段と、を備える記憶装置内容整理システム。

【請求項11】 請求項10記載の記憶装置内容整理システムであって、前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムに関してユーザが設定する環境設定を保存するための環境設定ファイルと、前記アプリケーションプログラムによって作成されたデータファイルとを含む、記憶装置内容整理システム。

【請求項12】 請求項10または11記載の記憶装置内容整理システムであって、前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムのインストールの時刻以降に作成または更新されたファイルを含む、記憶装置内容整理システム。

【請求項13】 コンピュータシステムに備えられた外

部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納されたファイルを実行するためのアイコンの1つを表示装置の画面上で選択するためのアイコン選択手段と、

前記アイコン選択手段で選択されたアイコンに対応して登録されているコマンドラインを解析して、前記コマンドラインがアプリケーションプログラムの実行ファイル以外のファイルを示す文字列を含むか否かを調べ、実行ファイル以外のファイルを示す文字列を含む場合には、前記実行ファイル以外のファイルを削除対象メインファイルとして特定するとともに、前記コマンドラインが実行ファイル以外のファイルを示す文字列を含まない場合には、前記実行ファイルを削除対象メインファイルとして特定する削除対象メインファイル特定手段と、前記削除対象メインファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする記憶装置内容整理システム。

【請求項14】 請求項13記載の記憶装置内容整理システムであって、さらに、前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示する検索手段を備える、記憶装置内容整理システム。

【請求項15】 請求項13または14記載の記憶装置内容整理システムであって、

前記検索手段は、前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合に、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする手段を備える、記憶装置内容整理システム。

【請求項16】 請求項15記載の記憶装置内容整理システムであって、前記削除候補となる情報は、WIN.INI ファイルおよび／またはレジストリファイル内の関連付け情報と、WIN.INI ファイル内のアプリケーションセクションと、前記レジストリファイル内のOLE 情報と、アイコンの登録情報とを含む、記憶装置内容整理システム。

【請求項17】 請求項15または16記載の記憶装置内容整理システムであって、さらに、前記削除候補となる情報を含むクリーン情報ファイルを作成する手段を備える、記憶装置内容整理システム。

【請求項18】 請求項14ないし17のいずれかに記載の記憶装置内容整理システムであって、前記削除候補のファイルの1つを内容表示対象ファイルとして選択する表示対象ファイル選択手段と、前記内容表示対象ファイルがアプリケーションプログラ

ムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示するファイル内容表示手段、を備える記憶装置内容整理システム。

【請求項19】 請求項14ないし18のいずれかに記載の記憶装置内容整理システムであって、さらに、前記検索手段によって検索された削除候補の中から、削除対象を選択する削除対象選択手段と、

前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによっては復元されないファイルを簡易バックアップ対象ファイルとして特定する簡易バックアップファイル特定手段と、

前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップするバックアップ実行手段と、を備える記憶装置内容整理システム。

【請求項20】 請求項19記載の記憶装置内容整理システムであって、

前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムに関してユーザが設定する環境設定を保存するための環境設定ファイルと、前記アプリケーションプログラムによって作成されたデータファイルとを含む、記憶装置内容整理システム。

【請求項21】 請求項19または20記載の記憶装置内容整理システムであって、

前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムのインストールの時刻以降に作成または更新されたファイルを含む、記憶装置内容整理システム。

【請求項22】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、

前記外部記憶装置に格納された1つのファイルを削除対象メインファイルとして特定する削除対象メインファイル特定手段と、

前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記削除対象メインファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示装置に表示する検索手段と、

前記削除候補の中から削除対象を選択する削除対象選択手段と、

前記削除対象メインファイルがアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによっては復元されないファイルを簡易バックアップ

対象ファイルとして特定する簡易バックアップファイル特定手段と、  
前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップするバックアップ実行手段と、  
前記削除対象メインファイルと前記削除対象選択手段によって選択された削除対象とを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする記憶装置内容整理システム。

【請求項23】 請求項22記載の記憶装置内容整理システムであって、  
前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムに関してユーザが設定する環境設定を保存するための環境設定ファイルと、前記アプリケーションプログラムによって作成されたデータファイルとを含む、記憶装置内容整理システム。

【請求項24】 請求項22または23記載の記憶装置内容整理システムであって、  
前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムのインストールの時刻以降に作成または更新されたファイルを含む、記憶装置内容整理システム。

【請求項25】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、  
前記外部記憶装置に格納された複数のファイルのリストを表示装置に表示するためのファイルリスト表示手段と、  
前記リストに表示されたファイルの1つを選択するファイル選択手段と、  
選択されたファイルがアプリケーションプログラムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示するファイル内容表示手段と、  
前記ファイルリストに表示された複数のファイルの中から削除の対象となるファイルを選択する削除対象選択手段と、  
前記対象特定手段によって選択されたファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする記憶装置内容整理システム。

【請求項26】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a) 前記外部記憶装置に格納されているファイルを分類するための分類情報を入力する工程と、(b) 前記分類情報に応じて、前記外部記憶装置に格納されているファイルを複数のファイル群に分類するとともに、各ファイル群のデータ総量を表示装置に表示する工程と、(c) 表示された前記複数のファイル群の1つを選択する工程と、(d) 選択されたファイル群に属するファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする記憶装置内

容整理方法。

【請求項27】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a) 前記外部記憶装置に格納されたファイルを実行するためのアイコンの1つを表示装置の画面上で選択するための工程と、(b) 選択されたアイコンに対応して登録されているコマンドラインを解析して、前記コマンドラインがアプリケーションプログラムの実行ファイル以外のファイルを示す文字列を含むか否かを調べ、実行ファイル以外のファイルを示す文字列を含む場合には、前記実行ファイル以外のファイルを削除対象メインファイルとして特定するとともに、前記コマンドラインが実行ファイル以外のファイルを示す文字列を含まない場合には、前記実行ファイルを削除対象メインファイルとして特定する工程と、  
(c) 前記削除対象メインファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする記憶装置内容整理方法。

【請求項28】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a) 前記外部記憶装置に格納された1つのファイルを削除対象メインファイルとして特定する工程と、(b) 前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記削除対象メインファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示装置に表示する工程と、(d) 前記削除候補の中から削除対象を選択する工程と、(e) 前記削除対象メインファイルがアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによっては復元されないファイルを簡易バックアップ対象ファイルとして特定する工程と、(f) 前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップする工程と、  
(g) 前記削除対象メインファイルと前記工程(d)で選択された削除対象とを前記外部記憶装置から削除する工程と、を備えることを特徴とする記憶装置内容整理方法。

【請求項29】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a) 前記外部記憶装置に格納された複数のファイルのリストを表示装置に表示する工程と、(b) 前記リストに表示されたファイルの1つを選択する工程と、(c) 選択されたファイルがアプリケーションプログラムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示する工程と、(d) 前記ファイルリス

トに表示された複数のファイルの中から削除の対象となるファイルを選択する工程と、(e)選択されたファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする記憶装置内容整理方法。

【請求項30】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、  
処理対象となるアプリケーションプログラムを選択するためのアプリケーション選択手段と、  
前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする検索手段と、  
前記削除候補を圧縮して書庫ファイルを作成する書庫化手段と、  
前記削除候補を前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする記憶装置内容整理システム。

【請求項31】 請求項30記載の記憶装置内容整理システムであって、さらに、  
前記削除候補となる情報を含むクリーン情報ファイルを作成する手段を備え、前記書庫化手段は、前記クリーン情報ファイルを含むように前記書庫ファイルを作成する手段を備える、記憶装置内容整理システム。

【請求項32】 請求項30または31記載の記憶装置内容整理システムであって、さらに、  
前記アプリケーションプログラムに対応するアイコンを、前記書庫ファイルを解凍する際に使用される解凍用アイコンに置き換えるアイコン変更手段を備える、記憶装置内容整理システム。

【請求項33】 請求項30ないし32のいずれかに記載の記憶装置内容整理システムであって、さらに、  
前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび／または情報を復元することによって、前記アプリケーションプログラムを前記削除候補の削除の前の状態に復元する解凍手段を備える、記憶装置内容整理システム。

【請求項34】 請求項33記載の記憶装置内容整理システムであって、  
前記解凍手段は、  
解凍されたアプリケーションプログラムの実行が終了する際に、前記削除候補のファイルおよび情報の中で、前記アプリケーションプログラムの実行中に変更されたものを再度圧縮して前記書庫ファイルを更新する書庫ファイル更新手段を備える、記憶装置内容整理システム。

【請求項35】 請求項33または34記載の記憶装置内容整理システムであって、さらに、

書庫化されるアプリケーションプログラムを含む書庫ファイルを示す第1の情報と、前記アプリケーションで作成されるドキュメントファイルの拡張子を示す第2の情報と、前記書庫ファイルが解凍された後に前記アプリケーションプログラムの実行ファイルを起動するためのコマンドラインを示す第3の情報とを含む関連付け情報を作成する手段を備え、

前記解凍手段は、

前記拡張子を有するドキュメントファイルの起動が指定された時に、前記関連付け情報に従って、前記アプリケーションプログラムを含む書庫ファイルを解凍するとともに、解凍された前記実行ファイルを起動する手段を備える、記憶装置内容整理システム。

【請求項36】 請求項35記載の記憶装置内容整理システムであって、  
前記関連付け情報の少なくとも一部は、オペレーティングシステム用のレジストリファイル内に格納される、記憶装置内容整理システム。

【請求項37】 請求項36記載の記憶装置内容整理システムであって、  
少なくとも前記第1と第2の情報が、前記レジストリファイル内に格納される、記憶装置内容整理システム。

【請求項38】 請求項35ないし37のいずれかに記載の記憶装置内容整理システムであって、  
前記関連付け情報の一部は、前記アプリケーションプログラムを含む書庫ファイル内に格納される、記憶装置内容整理システム。

【請求項39】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを書庫化するための書庫化システムであって、  
処理対象となるアプリケーションプログラムを選択するためのアプリケーション選択手段と、

前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを書庫化候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を書庫化候補とする検索手段と、  
前記書庫化候補を圧縮して書庫ファイルを作成する書庫化手段と、

前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび／または情報を復元することによって、前記アプリケーションプログラムを前記書庫化候補の圧縮の前の状態に復元する解凍手段と、を備えることを特徴とする書庫化システム。

【請求項40】 請求項39記載の書庫化システムであって、  
前記書庫化ファイルは、前記アプリケーションの設定状態を示す設定情報を含み、

前記解凍手段は、

前記設定情報に従って、前記アプリケーションプログラムの設定状態を前記書庫化候補の圧縮の前の状態に復元する手段を含む、書庫化システム。

【請求項41】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a) 処理対象となるアプリケーションプログラムを選択する工程と、(b) 前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする工程と、(c) 前記削除候補を圧縮して書庫ファイルを作成する工程と、(d) 前記削除候補を前記外部記憶装置から削除する工程と、を備えることを特徴とする記憶装置内容整理方法。

【請求項42】 コンピュータシステムに備えられた外部記憶装置に格納されているファイルを書庫化するための書庫化方法であって、(a) 処理対象となるアプリケーションプログラムを選択する工程と、(b) 前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを書庫化候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を書庫化候補とする工程と、(c) 前記書庫化候補を圧縮して書庫ファイルを作成する工程と、(d) 前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび／または情報を復元することによって、前記アプリケーションプログラムを前記書庫化候補の圧縮の前の状態に復元する工程と、を備えることを特徴とする書庫化方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムおよびその方法に関する。

【0002】

【従来の技術】コンピュータシステムの外部記憶装置としては、ハードディスク、光磁気ディスク等の書込可能な種々の記憶装置が使用されている。近年では、これらの外部記憶装置の大容量化が進んでおり、数百メガバイト、数ギガバイトの容量を持つ装置も普及している。

【0003】コンピュータシステムに多数のアプリケーションプログラムをインストールしたり、また、画像データなどを保存したりすると外部記憶装置に格納されるデータ量が膨大になってゆく。外部記憶装置の容量が大

きくても、格納すべきデータ量も膨大になるので、次第に外部記憶装置の空き容量の不足に悩む場合が多い。

【0004】従来は、このような場合に、外部記憶装置に格納された個々のファイルを指定して外部記憶装置から削除するような記憶装置内容整理用のアプリケーションプログラムが用いられていた。

【0005】

【発明が解決しようとする課題】しかし、例えば外部記憶装置からアプリケーションプログラムを削除するような場合には、そのアプリケーションプログラムに関連するファイルが極めて多数にのぼるので、個々のファイルを指定することによってアプリケーションプログラムに関連したファイルをきれいに削除するのは困難であった。また、個々のファイルを指定して削除するのは多大の手間と時間を要し、また、各ファイルに関する詳細な知識も必要であった。従って、従来はファイルの削除によって外部記憶装置の空き容量を効率的に増加させることが困難であった。

【0006】この発明は、従来技術における上述の課題を解決するためになされたものであり、削除すべきファイルを容易に選択でき、ファイルの削除によって外部記憶装置の空き容量を容易に増加させることができるようにすることを目的とする。

【0007】

【課題を解決するための手段およびその作用・効果】上述の課題を解決するため、第1の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納されているファイルを分類するための分類情報を入力するための分類情報入力手段と、前記分類情報に応じて、前記外部記憶装置に格納されているファイルを複数のファイル群に分類するとともに、各ファイル群のデータ総量を表示装置に表示する分類手段と、表示された前記複数のファイル群の1つを選択するためのファイル群選択手段と、前記選択手段によって選択されたファイル群に属するファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする。

【0008】ユーザが分類情報を入力すると、分類手段がその分類情報に応じて外部記憶装置内に格納されたファイルを複数のファイル群に分類する。そして、ユーザが分類された複数のファイル群の1つを選択すると、削除実行手段が選択されたファイル群に属するファイルを削除する。従って、簡単な操作で、分類された特定のファイル群を容易に削除でき、外部記憶装置の空き容量を増加させることができる。

【0009】上記第2の発明において、前記分類情報は、ファイルの拡張子別にファイルを分類する旨の情報を含むようにすることが好ましい。

【0010】こうすれば、特定の拡張子を有するファイ

ル群を選択して削除できる。

【0011】また、前記分類情報は、ディレクトリ単位でファイルを分類する旨の情報を含み、前記削除実行手段は、前記選択手段で選択されたディレクトリおよび該ディレクトリに含まれるファイルを削除する手段を備えるようにすることが好ましい。

【0012】こうすれば、特定のディレクトリ内に含まれているファイル群を選択して削除できる。

【0013】上記第1の発明は、さらに、前記外部記憶装置内の各ファイルへのアクセス回数とアクセス日時とを含むアクセス履歴を記録する履歴記録手段を含み、前記分類情報は、前記履歴記録手段によって記録されたアクセス履歴に関する分類条件を示す情報を含むことが好ましい。

【0014】こうすれば、特定のアクセス履歴の条件を満たすファイル群を選択して削除できる。

【0015】上記第1の発明は、さらに、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合には、前記実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示する検索手段を備えることが好ましい。

【0016】こうすれば、実行ファイルの実行時にのみ使用される可能性のあるファイルが削除候補として表示されるので、ユーザがこれを確認して不要なファイルを削除することができる。

【0017】また、前記検索手段は、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合に、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする手段を備えることが好ましい。

【0018】こうすれば、所定のファイル内の不要な情報を削除することができる。

【0019】また、前記削除候補となる情報は、WIN.INI ファイルおよび/またはレジストリファイル内の関連付け情報と、WIN.INI ファイル内のアプリケーションセクションと、前記レジストリファイル内のOLE 情報と、アイコンの登録情報とを含むことが好ましい。

【0020】さらに、前記削除候補となる情報を含むクリーン情報ファイルを作成する手段を備えることが好ましい。

【0021】こうすれば、削除候補となった情報を削除した後に、クリーン情報ファイルを用いてその情報を復元することができる。

【0022】また、上記第1の発明は、前記削除候補のファイルの1つを内容表示対象ファイルとして選択する表示対象ファイル選択手段と、前記内容表示対象ファイルがアプリケーションプログラムによって作成されたデ

ータファイルである場合に、前記データファイルの内容を前記表示装置に表示するファイル内容表示手段、を備えることが好ましい。

【0023】こうすれば、ユーザがデータファイルの内容を確認して、そのファイルを削除するか否かを判断することができる。

【0024】上記第1の発明は、さらに、前記検索手段によって検索された削除候補の中から、削除対象を選択する削除対象選択手段と、前記ファイル群選択手段によって選択されたファイル群がアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによって復元されないファイルを簡易バックアップ対象ファイルとして特定する簡易バックアップファイル特定手段と、前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップするバックアップ実行手段と、を備えることが好ましい。

【0025】こうすれば、アプリケーションプログラムの復元に必要なバックアップファイルの容量を小さくできる。

【0026】また、前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムに関してユーザが設定する環境設定を保存するための環境設定ファイルと、前記アプリケーションプログラムによって作成されたデータファイルとを含むようにすることが好ましい。

【0027】これらのファイルは、アプリケーションプログラムのインストール後に、ユーザによって作成または変更された可能性があるので、これらをバックアップしておけば、アプリケーションプログラムに関するファイルを削除前の状態に復元することができる。

【0028】また、前記簡易バックアップ対象ファイルは、前記アプリケーションプログラムのインストールの時刻以降に作成または更新されたファイルを含むようにすることが好ましい。

【0029】このようなファイルは、アプリケーションプログラムのインストール後に、ユーザによって作成または変更されたものなので、これらをバックアップしておけば、アプリケーションプログラムに関するファイルを削除前の状態に復元することができる。

【0030】第2の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納されたファイルを実行するためのアイコンの1つを表示装置の画面上で選択するためのアイコン選択手段と、前記アイコン選択手段で選択されたアイコンに対応して登録されているコマンドラインを解析して、前記コマンドラインがアプリケーションプログラムの実行ファイル以外のファイルを示す文字列を含むか否かを調べ、実行ファイル以外のファイルを示す文字列を



含む場合には、前記実行ファイル以外のファイルを削除対象メインファイルとして特定するとともに、前記コマンドラインが実行ファイル以外のファイルを示す文字列を含まない場合には、前記実行ファイルを削除対象メインファイルとして特定する削除対象メインファイル特定手段と、前記削除対象メインファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする。

【0031】削除対象メインファイル特定手段は、アイコンとして登録されている実体的なファイルを削除対象メインファイルと特定するので、削除実行手段がそのファイルを削除することができる。

【0032】また、第3の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納された1つのファイルを削除対象メインファイルとして特定する削除対象メインファイル特定手段と、前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記削除対象メインファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示装置に表示する検索手段と、前記削除候補の中から削除対象を選択する削除対象選択手段と、前記削除対象メインファイルがアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファイルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによって復元されないファイルを簡易バックアップ対象ファイルとして特定する簡易バックアップファイル特定手段と、前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップするバックアップ実行手段と、前記削除対象メインファイルと前記削除対象選択手段によって選択された削除対象とを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする。

【0033】簡易バックアップファイル特定手段は、アプリケーションプログラムの復元に必要なバックアップファイルの数を削減するので、バックアップファイルの容量を小さくできる。

【0034】第4の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、前記外部記憶装置に格納された複数のファイルのリストを表示装置に表示するためのファイルリスト表示手段と、前記リストに表示されたファイルの1つを選択するファイル選択手段と、選択されたファイルがアプリケーションプログラムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示するファイル内容表示手段と、前記ファイルリストに表示

された複数のファイルの中から削除の対象となるファイルを選択する削除対象選択手段と、前記対象特定手段によって選択されたファイルを前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする。

【0035】ファイル内容表示手段によって表示されたデータファイルの内容をユーザが確認して、そのファイルを削除するか否かを判断することができる。

【0036】第5の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a)前記外部記憶装置に格納されているファイルを分類するための分類情報を入力する工程と、(b)前記分類情報に応じて、前記外部記憶装置に格納されているファイルを複数のファイル群に分類するとともに、各ファイル群のデータ総量を表示装置に表示する工程と、(c)表示された前記複数のファイル群の1つを選択する工程と、

(d)選択されたファイル群に属するファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする。

【0037】また、第6の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、

(a)前記外部記憶装置に格納されたファイルを実行するためのアイコンの1つを表示装置の画面上で選択するための工程と、(b)選択されたアイコンに対応して登録されているコマンドラインを解析して、前記コマンドラインがアプリケーションプログラムの実行ファイル以外のファイルを示す文字列を含むか否かを調べ、実行ファイル以外のファイルを示す文字列を含む場合には、前記実行ファイル以外のファイルを削除対象メインファイルとして特定するとともに、前記コマンドラインが実行ファイル以外のファイルを示す文字列を含まない場合には、前記実行ファイルを削除対象メインファイルとして特定する工程と、(c)前記削除対象メインファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする。

【0038】第7の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a)前記外部記憶装置に格納された1つのファイルを削除対象メインファイルとして特定する工程と、(b)前記削除対象メインファイルがアプリケーションプログラムの実行ファイルである場合には、前記削除対象メインファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補として表示装置に表示する工程と、(d)前記削除候補の中から削除対象を選択する工程と、(e)前記削除対象メインファイルがアプリケーションプログラムの実行ファイルを含む場合には、前記削除対象として選択されたファ

イルの中で前記アプリケーションプログラムをコンピュータシステムに再インストールすることによって復元されないファイルを簡易バックアップ対象ファイルとして特定する工程と、(f)前記簡易バックアップ対象ファイルを圧縮データファイルとしてバックアップする工程と、(g)前記削除対象メインファイルと前記工程(d)で選択された削除対象とを前記外部記憶装置から削除する工程と、を備えることを特徴とする。

【0039】第8の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a)前記外部記憶装置に格納された複数のファイルのリストを表示装置に表示する工程と、(b)前記リストに表示されたファイルの1つを選択する工程と、(c)選択されたファイルがアプリケーションプログラムによって作成されたデータファイルである場合に、前記データファイルの内容を前記表示装置に表示する工程と、(d)前記ファイルリストに表示された複数のファイルの中から削除の対象となるファイルを選択する工程と、(e)選択されたファイルを前記外部記憶装置から削除する工程と、を備えることを特徴とする。

【0040】第9の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理システムであって、処理対象となるアプリケーションプログラムを選択するためのアプリケーション選択手段と、前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする検索手段と、前記削除候補を圧縮して書庫ファイルを作成する書庫化手段と、前記削除候補を前記外部記憶装置から削除する削除実行手段と、を備えることを特徴とする。

【0041】検索手段によって削除候補とされたファイルと情報を書庫化手段が書庫化するので、書庫ファイルを解凍することによって削除されたファイルと情報を元の通り復元することができる。

【0042】上記第9の発明は、さらに、前記削除候補となる情報を含むクリーン情報ファイルを作成する手段を備え、前記書庫化手段は、前記クリーン情報ファイルを含むように前記書庫ファイルを作成する手段を備えることが好ましい。

【0043】こうすれば、書庫ファイルの解凍時にクリーン情報ファイルを用いて元のファイルや情報を容易に復元することができる。

【0044】また、第9の発明は、さらに、前記アプリケーションプログラムに対応するアイコンを、前記書庫ファイルを解凍する際に使用される解凍用アイコンに置

き換えるアイコン変更手段を備えるようにすることが好ましい。

【0045】こうすれば、ユーザが解凍用アイコンを指定して容易に書庫ファイルの解凍を指示することができる。

【0046】第9の発明は、さらに、前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび/または情報を復元することによって、前記アプリケーションプログラムを前記削除候補の削除の前の状態に復元する解凍手段を備えることが好ましい。

【0047】こうすれば、アプリケーションプログラムが元の状態に復元されるので、ユーザは解凍されていたことをほとんど意識せずに、そのアプリケーションを利用することができる。

【0048】第9の発明の前記解凍手段は、解凍されたアプリケーションプログラムの実行が終了する際に、前記削除候補のファイルおよび情報の中で、前記アプリケーションプログラムの実行中に変更されたものを再度圧縮して前記書庫ファイルを更新する書庫ファイル更新手段を備えることが好ましい。

【0049】こうすれば、書庫ファイルの内容を、そのアプリケーションプログラムの最新の状態に更新することができる。

【0050】第9の発明は、さらに、書庫化されるアプリケーションプログラムを含む書庫ファイルを示す第1の情報と、前記アプリケーションで作成されるドキュメントファイルの拡張子を示す第2の情報と、前記書庫ファイルが解凍された後に前記アプリケーションプログラムの実行ファイルを起動するためのコマンドラインを示す第3の情報とを含む関連付け情報を作成する手段を備え、前記解凍手段は、前記拡張子を有するドキュメントファイルの起動が指定された時に、前記関連付け情報に従って、前記アプリケーションプログラムを含む書庫ファイルを解凍するとともに、解凍された前記実行ファイルを起動する手段を備える、ことが好ましい。

【0051】こうすれば、ユーザは、プログラムが圧縮されていることを意識せずに、ドキュメントファイルを利用できる。

【0052】前記関連付け情報の少なくとも一部は、オペレーティングシステム用のレジストリファイル内に格納されるようにしてもよい。もちろん、関連付け情報の全部をレジストリファイル内に格納するようによい。また、少なくとも前記第1と第2の情報が、前記レジストリファイル内に格納されるようにしてもよい。さらに、前記関連付け情報の一部が、前記アプリケーションプログラムを含む書庫ファイル内に格納されるようにしてもよい。

【0053】第10の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを書

庫化するための書庫化システムであって、処理対象となるアプリケーションプログラムを選択するためのアプリケーション選択手段と、前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを書庫化候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を書庫化候補とする検索手段と、前記書庫化候補を圧縮して書庫ファイルを作成する書庫化手段と、前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび／または情報を復元することによって、前記アプリケーションプログラムを前記書庫化候補の圧縮の前の状態に復元する解凍手段と、を備えることを特徴とする。

【0054】このような書庫化システムを例えば2つのコンピュータにおいて動作させれば、2つのコンピュータ間において、アプリケーションプログラムを書庫ファイルの形で容易に転送することができる。この場合に、転送先のコンピュータにおけるアプリケーションの状態を、転送元のコンピュータにおける状態と同じにすることができる。

【0055】上記第10の発明において、前記書庫化ファイルは、前記アプリケーションの設定状態を示す設定情報を含み、前記解凍手段は、前記設定情報に従って、前記アプリケーションプログラムの設定状態を前記書庫化候補の圧縮の前の状態に復元する手段を含むようにすることが好ましい。

【0056】こうすれば、例えば2つのコンピュータ間でアプリケーションプログラムを転送した場合にも、その設定状態を同一にすることができる。

【0057】第11の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを整理するための記憶装置内容整理方法であって、(a)処理対象となるアプリケーションプログラムを選択する工程と、(b)前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを削除候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を削除候補とする工程と、(c)前記削除候補を圧縮して書庫ファイルを作成する工程と、(d)前記削除候補を前記外部記憶装置から削除する工程と、を備えることを特徴とする。

【0058】第12の発明は、コンピュータシステムに備えられた外部記憶装置に格納されているファイルを書庫化するための書庫化方法であって、(a)処理対象となるアプリケーションプログラムを選択する工程と、(b)前記アプリケーションプログラムの実行ファイルの実行に伴って使用され、かつ、他の実行ファイルの実

行に使用されない他のファイルを前記外部記憶装置の内部から検索して、検索されたファイルを書庫化候補とするとともに、前記実行ファイルに関して所定のファイル内に含まれている情報を書庫化候補とする工程と、

(c)前記書庫化候補を圧縮して書庫ファイルを作成する工程と、(d)前記書庫ファイルの解凍が指示された時に、前記書庫ファイルを解凍するとともに、前記書庫ファイルに含まれていたファイルおよび／または情報を復元することによって、前記アプリケーションプログラムを前記書庫化候補の圧縮の前の状態に復元する工程と、を備えることを特徴とする。

【0059】

【発明の他の態様】この発明は、以下のような他の態様も含んでいる。第1の態様は、コンピュータシステムのマイクロプロセッサによって実行されることによって上記の各発明の各種の手段や工程を実現するソフトウェアプログラムを格納した記憶媒体（特に携帯型記憶媒体）である。

【0060】第2の態様は、コンピュータシステムのマイクロプロセッサによって実行されることによって上記の各発明の各種の手段や工程を実現するソフトウェアプログラムを通信回線を介して供給するプログラム供給装置である。

【0061】

【発明の実施の形態】

A. 装置の構成：次に、本発明の実施の形態を実施例に基づき説明する。図1は、この発明の一実施例を適用するコンピュータシステムのハードウェア構成を示すブロック図である。このコンピュータ10は、ローカルバス22に接続された演算処理部20、ローカルバス22を外部バスの一つであるPCIバス32に接続するPCIブリッジ30、PCIバス32を介して演算処理部20のCPU21等によりアクセスを受けるコントローラ部40、各種のI/O装置等を制御する機器が低速の外部バスであるISAバス42に接続されたI/O部60、および周辺機器であるキーボード72、スピーカ74、CRT76などから構成されている。

【0062】演算処理部20は、中央演算処理装置としてのCPU21、キャッシュメモリ23、そのキャッシュコントローラ24およびメインメモリ25から構成されている。PCIブリッジ30は、高速のPCIバス32を制御する機能を備えたコントローラである。CPU21が扱うメモリ空間は、CPU21の内部に用意された各種レジスタにより、実際の物理アドレスより広い論理アドレスに拡張されている。

【0063】コントローラ部40は、モニタ(CRT)76への画像の表示を司るグラフィックスコントローラ（以下、VGAと呼ぶ）44、接続されるSCSI機器とのデータ転送を司るSCSIコントローラ46、PCIバス32と下位のISAバスとのインタフェースを司

るPCI-ISAブリッジ48から構成されている。VGA44は、CRT76に対して、640×480ドット、16色表示が可能である。なお、表示用のフォントを記憶したキャラクタジェネレータや所定のコマンドを受け取って所定の図形を描画するグラフィックコントローラ、さらには描画画像を記憶するビデオメモリ等は、このVGA44に実装されているが、これらの構成は周知のものなので、図1では省略されている。

【0064】PCI-ISAブリッジ48を介して接続されたISAバス42は、各種のI/O機器が接続される入出力制御用のバスであり、DMAコントローラ（以下単にDMAと呼ぶ）50、リアルタイムクロック（RTC）52、複合I/Oポート54、サウンドI/O56、キーボード72およびマウス73とのインタフェースを司るキーボードインタフェース（以下KEYと呼ぶ）64、優先順位を有する割り込み制御を行なう割り込みコントローラ（以下PICと呼ぶ）66、各種の時間カウントやビーブ音を発生するタイマ68等から構成されている。なお、ISAバス42には、拡張ボードが実装可能なISAスロット62が接続されている。

【0065】複合I/Oポート54には、パラレル出力、シリアル出力の他、フロッピディスク装置82やハードディスク84を制御する信号を入出力するポートが用意されている。また、パラレル入出力には、パラレルポート86を介してプリンタ88が、シリアル入出力には、シリアルポート90を介してモデム92が、各々接続されている。また、サウンドI/O56には、上述したスピーカ74の他、マイクロフォン96が接続可能とされている。これらの構成の他、コンピュータ10では、標準化されたI/Oチャンネルが用意されることも多いが、本実施例では図示および説明は省略する。

【0066】なお、以下ではハードディスク84内のファイルの整理を行なう場合について説明する。但し、本発明は、ハードディスクや光磁気ディスクなどの種々のタイプの書込可能な外部記憶装置に格納されたファイルの整理に適用することができる。

【0067】図2は、ハードディスク84内のファイルの整理を行なうプログラム群の全体の機能を示す機能ブロック図である。このプログラム群が記憶装置内容整理システムおよび記憶装置内容整理方法を実現している。これらのプログラム群は、フロッピディスクやCD-ROM等の携帯型記憶媒体（可搬型記憶媒体）からコンピュータシステムのハードディスク84やメインメモリ25等の記憶装置に転送される。あるいは、これらのプログラム群を、通信回線を介してプログラム供給装置からコンピュータシステムに供給するようにしてもよい。実行時には、記憶装置からこれらのプログラム群が読み出され、CPU21によって実行される。なお、以下の説明においては、「機能」と「手段」とが同義語として使用される場合がある。

【0068】ディスク内容の整理に関連するプログラム群は、ファインダ部202と、クリーナ部204と、リソース監視部206と、アークファイラ部208と、レスキュー部210と、履歴記録部212とを備えている。

【0069】履歴記録部212は、アクセス監視部220と、履歴ログ222と、履歴データベース更新部224と、履歴データベース226とを含んでいる。なお、要素202、204、206、208、210の機能は1つのプログラムにまとめられており、アクセス監視部220と履歴データベース更新部224は、これと独立したそれぞれ別個のプログラムである。また、アクセス監視部220は常駐プログラム（TSR）であり、メインメモリ25内に常駐している。履歴ログ222と履歴データベース226は、実際にはハードディスク84内に格納されている。履歴記録部212は、本発明における履歴記憶手段に相当する。

【0070】以下では、まず図2の各部の機能の概要について順次説明し、その後、ファイル削除の手順について説明する。この実施例では、オペレーティングシステムとしてMS-Windowsシステム（マイクロソフト社の商標）を使用した場合について説明する。

【0071】B. ファインダ部202の機能の概要：ファインダ部202は、ハードディスク84内のファイル管理用ツールである。ファインダ部202の主要な機能は、ハードディスク84内に格納されているファイルを各種のモードで一覧表示する機能である。ユーザは、これらの一覧表示を利用してハードディスク84の使用状況を把握し、不要なファイルを検索することができる。ユーザは、一覧表示の中から不要なファイルを選択して削除することもできる。ファインダ部202による一覧表示の機能としては、以下に説明する「ドライブ別一覧」、「ディレクトリ別一覧」、「拡張子別一覧」、「アクセス履歴を含むファイル一覧」、「カレンダー表示」がある。

【0072】（1）ドライブ別一覧：この機能では、コンピュータシステムの各ディスクドライブのドライブ名（A、B、C等）、ボリューム名、全容量、使用容量、空き容量、使用率（＝使用容量／全容量）等が一覧表示される。ユーザは、この一覧表示によって、どのディスクドライブのファイルを整理すべきかを判断することができる。

【0073】（2）ディレクトリ別一覧：この機能は、各ディレクトリがどれくらいハードディスク84を占有しているかを一覧表示する機能である。図3は、ファインダ部202によるディレクトリ別一覧の画面表示の一例を示す図である。画面の左端にある4つのボタン「ドライブ」、「ディレクトリ」、「拡張子」、「履歴」、「カレンダー」は、それぞれ「ドライブ別一覧」、「ディレクトリ別一覧」、「拡張子別一覧」、「アクセス履歴

を含むファイル一覧」、「カレンダー表示」の機能を選択するための一覧表示選択ボタンである。ディレクトリ別一覧では、各ディレクトリの名称、各ディレクトリ内に含まれているファイル数、ディレクトリの使用容量、ディスクドライブ内での各ディレクトリの利用率等が表示される。ユーザは、この一覧表示から、どのディレクトリを削除すればディスクドライブの空き容量を増加させることができるかを容易に判断することができる。また、一覧表示されたディレクトリの1つをマウス73で選択し、画面上にある「削除」のボタンを押せば、そのディレクトリとディレクトリ内の全ファイルを削除することができる。

【0074】(3) 拡張子別一覧：この機能は、各拡張子のファイルがどれくらいハードディスク84を占有しているかを一覧表示する機能である。「拡張子」は、ファイル名に付加されている3文字の記号であり、ファイルの種類によってそれぞれ異なる拡張子が付加されている。図4は、ファインダ部202による拡張子別一覧の画面表示の一例を示す図である。拡張子別一覧では、各拡張子に対応するアプリケーション名（アプリケーションに関連付けられないファイルはそのファイルの種類）、各拡張子を有するファイル数、各拡張子を有するファイルの使用容量、ディスクドライブ内での利用率等が表示される。ユーザは、この一覧表示から、どの拡張子のファイルを削除すればディスクドライブの空き容量を増加させることができるかを容易に判断することができる。また、一覧表示された拡張子をマウス73で選択し、画面上にある「削除」のボタンを押せば、その拡張子を有する全ファイルを削除することができる。

【0075】以上のように、ファインダ部202は、「ドライブ別」、「ディレクトリ別」、「拡張子別」等の分類情報に応じて、ハードディスク84に格納されているファイルを複数のファイル群に分類し、分類されたファイル群を一覧表示する分類手段としての機能を有している。図3や図4に示す画面の左端に表示された「ドライブ」、「ディレクトリ」、「拡張子」のボタンとマウス73は、分類情報入力手段としての機能を実現している。また、分類されて一覧表示されたファイル群を選択する際のマウス73はファイル群選択手段としての機能を実現している。ファインダ部202はさらに、選択されたファイル群を削除する削除実行手段としての機能も有している。

【0076】(4) アクセス履歴を含むファイル一覧：図5は、アクセス履歴を含むファイル一覧の画面表示の一例を示す図である。各ファイルのアクセス履歴は、履歴記録部212（図2）内の履歴データベース226に記録されている。ユーザが図5の「履歴」のボタンをマウス73で押すと、図5に示すように個々のファイルのパス名（ドライブ名とディレクトリ名とファイル名とで構成される）、アクセス数、最終アクセス日時、利用

度、登録日、更新日時等が表示される。ここで、「登録日」とは、そのファイルが履歴データベース226に最初に登録された日時を示す。また、「更新日時」とは、そのファイルが更新された日時を示す。「利用度」は、アクセスされた日時とアクセス回数に応じて例えば図6のように分類される。アクセス履歴の一覧を使用すれば、あまり使用していないファイルを容易に探すことができる。なお、ユーザは、図6に示す利用度の分類条件を変更することができるので、特定の分類条件を満足するファイルを容易に検索することができる。

【0077】ところで、図5のようなファイルの一覧表示であり利用されていないファイルを発見した際に、そのファイルの内容を確認して削除するか否かを判断したい場合がある。この場合には、図5の上部にある「詳細」ボタンをマウス73で押すと、図7に示すファイルの詳細情報が画面に表示される。ファイルの詳細情報としては、パス名を含むファイル名、更新日時、ファイルサイズ、アクセス回数、最終アクセス日時、および、ファイルの概要が表示される。ファイルの概要としては、そのファイルがアプリケーションプログラムの実行ファイルの場合には、そのアプリケーションプログラムの名前とアイコンとが表示される。また、アプリケーションプログラムで作成されたデータファイルである場合には、その旨が表示される。

【0078】詳細情報が表示されたファイルの内容を表示したい場合には、図7の画面の右端にある「表示」のボタンをマウス73で押せばよい。こうすると、そのファイルの内容を画面に表示することができる。例えば、そのファイルがワードプロセッサ、グラフィックス、表計算プログラム等のアプリケーションプログラムで作成されたデータファイルである場合には、そのアプリケーションプログラムを立上げることなく、その内容（文書、画像、スプレッドシート等）が画面上に表示される。この表示機能は、ファイル内容を画面に展開するプログラムを用いて実現することができる。このプログラムは、各アプリケーションプログラムのデータファイルのファイル構造を保存しており、そのファイル構造に従ってファイル内容を画面に展開する。ファインダ部202は、このようにファイルの内容を画面に表示できるので、個々のファイルを実際に削除してもよいか否かをユーザが容易に判断することができる。

【0079】なお、図5において、ファイルを一覧表示する機能は本発明におけるファイルリスト表示手段に相当し、この一覧表示の中からマウス73を用いて詳細表示を行なうファイルを選択する機能は本発明における表示対象ファイル選択手段に相当する。また、図7の「表示」ボタンを押してファイルの内容を表示する機能は、本発明におけるファイル内容表示手段に相当する。

【0080】(5) カレンダー表示：カレンダー表示では、カレンダーに類似した画面が表示され、その個々の日にア

クセスされたファイル名が示される。この機能を利用すれば、アクセス日時からファイルを検索することができる。

【0081】C. クリーナ部204の機能の概要：クリーナ部204は、以下に説明するように、アプリケーションプログラムの削除や、不要ファイルの削除、所定のファイル内の情報の削除等を行なう削除実行手段としての種々の機能を有している。

【0082】(1) アプリケーションのアンインストール機能：アプリケーションのアンインストール機能とは、アプリケーションを実行するための種々のファイルと情報をハードディスク84から削除する機能を言う。アプリケーションのアンインストールを行なうには、アプリケーションの実行ファイル（EXE ファイル）の削除の他に、そのアプリケーションのみに使用されているDLL ファイルの削除、アプリケーションのINI ファイル（設定ファイル）の削除、アイコンの削除、システムのファイル（WIN.INI やレジストリファイル等）に登録された情報の削除等の処理が必要となる。アプリケーションのアンインストール機能の詳細は後述する。

【0083】(2) INI ファイルのクリーン機能：INI ファイルのクリーン機能は、不要なINI ファイルの削除と、Windows システムのINI ファイルであるWIN.INI ファイル内の不要な記述の削除とを行なう機能である。INI ファイルとは、種々の環境設定を保存するための設定ファイルである。Windows システムに関連するINI ファイルとしては、Windows システム全般に関する設定を保存するWIN.INI や、デバイスドライバに関する設定等を保存するSYSTEM.INIなどがある。また、各アプリケーションもそれぞれ独自のINI ファイルを有していることが多い。

【0084】図8は、WIN.INI のファイル内容の一例を示す説明図である。一般に、INI ファイルはテキストファイルであり、図8に示すように、括弧〔 〕で囲まれたセクションに分けられている。また、各セクションには、等号「=」の前に置かれたキーと、等号の後に置かれた値との組み合わせで表現された種々の記述が含まれている。Windows セクションには、Windows システムの全般に関係する設定が登録されている。例えば「run」というキーは、Windows システムの起動時に起動するアプリケーションプログラムを設定するためのキーである。Desktop セクションには、画面の背景パターンや壁紙等の設定が登録されている。また、Extensionsセクションには、アプリケーションプログラムの実行ファイルと拡張子との関連付けが登録されている。例えば、「cal」という拡張子は、calendar.exeという実行ファイルに関連付けられている。

【0085】アプリケーションプログラムがインストールされると、そのアプリケーションプログラムに関する関連付けの記述がWIN.INI 内のExtensionsセクションに

追加されるのが普通である。図8の例では、「Disk 2.0」というアプリケーションプログラムの拡張子「dxt」と、その実行ファイルDXUTW.EXE との関連付けが追加されている。また、アプリケーションプログラムのインストールの際には、そのアプリケーション専用のセクションがWIN.INI 内に追加されることが多い。図8の例では、[DXUTW] というセクションが追加されている。アプリケーションプログラム専用のセクション名は、その実行ファイルのファイル名から拡張子を除いたものに設定されるのが普通である。

【0086】クリーナ部204のINI ファイルのクリーン機能を起動すると、ハードディスク84内のINI ファイルが一覧表示される。ユーザは一覧表示の中から不要なINI ファイルを選択して削除を実行することによって、不要なINI ファイルを容易に削除することができる。また、INI ファイルのクリーン機能の一部として、WIN.INI 内のセクションの記述を削除したり変更したりするINI ファイルの編集機能も備えられている。

【0087】(3) 重複ファイルのクリーン機能：重複ファイルのクリーン機能は、異なるディレクトリ内に格納されている同一名のファイルを表示する機能である。ユーザは、表示されたファイルの中で不要なファイルを選択して削除することができる。また、図7に示したような詳細情報の表示や、ファイルの内容表示機能を利用することによって、ファイルのアクセス履歴や内容を確認した後に、そのファイルを削除するか否かを決定することも可能である。

【0088】(4) 孤立ファイルのクリーン機能：孤立ファイルのクリーン機能は、参照されていないファイルを削除するための機能である。この機能を起動すると、実行ファイルから参照されていないDLL ファイル（後述する）や、プログラムマネージャから参照されていないグループファイル、アプリケーションのアンインストール後に残っているそのアプリケーションのINI ファイル等が一覧表示される。ユーザは、このようなファイルの一覧表示の中から不要なファイルを指定して、削除を実行する。

【0089】DLL ファイルとは、Windows システムのアプリケーションプログラムの実行ファイルに動的にリンクされる動的リンクライブラリ（Dynamic Link Library）と呼ばれているライブラリである。従来のアプリケーションではライブラリ部分は静的なリンクであり、アプリケーションプログラム作成時にライブラリがプログラムの一部に埋め込まれていた。一方、DLL ファイルでは、アプリケーションプログラム作成時には、ライブラリのファイル名と呼び出す関数名がプログラムに書き込まれるだけであり、ライブラリ部分は別ファイルとして提供される。そして、アプリケーションの実行ファイルは、実行時に初めてそのDLL ファイルと結合されるようになっている。Windows システムの場合には、そのオペ

レーティングシステムの各種の機能を利用するためのシステムコール自体がこのDLL ファイルとなっているため、すべてのアプリケーションプログラムが直接的、間接的にDLL ファイルを利用していることになる。アプリケーションのインストール時には、これらのDLL ファイルをアプリケーション自身のディレクトリにコピーするようになっているものが多い。しかし、他のプログラムとDLL ファイルを共有するような場合には、Windows システムのディレクトリにDLL ファイルをコピーする場合もある。このような場合には、どのアプリケーションがどのDLL ファイルを参照しているかは表向きわからないので、実行ファイルとDLL ファイルの参照関係を何らかの方法で調べる必要がある。なお、DLL ファイルの参照関係は、後述する参照DLL の解析機能を利用することによって調べることができる。

【0090】(5) Windowsシステムの不要ファイルクリーン機能：この機能は、スタンダードモード用のファイルや、利用しないユーティリティプログラムなどのWindows システムのファイルをカテゴリ別にファイル表示して、不要ファイルの削除を行なうための機能である。カテゴリとしては、ディスプレイドライバ、プリンタドライバ、フォント、IME（日本語入力プログラム）等がある。ユーザは、カテゴリ別のファイルの一覧表示の中から不要なファイルを指定して、削除を実行する。

【0091】(6) DOSシステムの不要ファイルクリーン機能：この機能は、MS-DOS（マイクロソフト社の商標）のファイルの中で不要なものの削除を行なうための機能である。Windows システムを主に利用している場合には、MS-DOSのシェルなどの使用しないシステムファイルを削除してもよい。この機能を起動すると、ファイル名とその概要が一覧表示されるので、ユーザが不要なファイルを指定して、削除を実行する。

【0092】(7) その他の不要ファイルクリーン機能：不要と考えられるものとしては、他に次のようなものがある。

- ①空きのディレクトリ。
- ②バックアップファイル（拡張子がbak, \$?, JAS等のもの）。なお、「?」は任意の一文字を示すワイルドカードである。
- ③サイズが0のファイル。
- ④テンポラリファイル（拡張子がtmp等のもの）。
- ⑤ディスク修復後のファイル（ファイル名がfile????\_dd, file????.chkなどのもの）。

【0093】クリーナ部204からその他の不要ファイルのクリーン機能を起動すると、これらのファイルの一覧が表示される。ユーザは、その中から不要なファイルを指定して、削除を実行する。

【0094】D. リソース監視部206の機能の概要：リソース監視部206は、システムリソースと外部記憶装置の使用状態を監視するための種々の機能を有してい

る。ここで、「システムリソース」とは、Windows メモリ、GDI（グラフィックス・デバイス・インタフェース）用メモリ、USERメモリ、コンベンショナルメモリ等のメモリ領域のことを言う。リソース監視部206の監視機能をオンに設定しておく、リソース監視部206が常にシステムリソースと各ディスクドライブの使用率を監視する。そして、システムリソースやディスクドライブの空き容量が所定値以下になった時に、画面表示や警告音などによってユーザに警告を発する。こうすれば、システムリソースの空き容量が低下した際に複数のアプリケーションの一部を終了するなどの対応を行なう、システムリソースの空き容量が過度に低下することを防止できるので、Windows システムのクラッシュを防止することができる。また、外部記憶装置の空き容量が少なくなり、ファイルの削除が必要となった状態を直に知ることができる。

【0095】E. アークファイラ部208の機能の概要：アークファイラ部208は、LZH, ZIP等の各種の圧縮データ形式のアーカイブファイル（書庫ファイル）の管理機能を有している。具体的には、ハードディスク84に格納されているファイルを圧縮して書庫ファイルを作成する機能と、書庫ファイルを解凍する機能とがある。また、ドラッグアンドドロップもサポートされており、アークファイラ部208のウィンドウと他のウィンドウとの間でファイルのアイコンを移動させるだけで、ファイルの圧縮や解凍を行なうことができる。

【0096】F. レスキュー部210の機能の概要：レスキュー部210は、Windows システムの動作に必要な各種のファイルの保存と復元を行なう機能を有している。ユーザが、正常に動作しているWindows システムをレスキュー部210に保存するように指示すると、その時点におけるWindows システムの起動に必要な各種のファイルがレスキュー部210によって保存される。保存されるファイルには、Windows システム用INI ファイル（WIN.COM, REG.DAT, WIN.INI, SYSTEM.INI 等）、メモリマネージャ、デバイスドライバ、CONFIG.SYS, AUTOEXEC.BAT等が含まれている。レスキュー部210は、MS-DOSシステムから起動することができ、Windows システムが起動しなくなったり正常に動作しなくなった場合に、保存していたファイルを復元して正常なWindows システムを構築することができる。

【0097】G. 履歴記録部212の機能の概要：履歴記録部212は、ハードディスク84に格納されたファイルがアクセスされたときに、そのアクセスを検出して、履歴データベース226にアクセス履歴として記録する機能を有している。ファイルのオープンが行なわれると、常駐プログラムであるアクセス監視部220（図2）がこれを検出して、履歴ログ222にそのファイルがアクセスされたことを記録する。履歴データベース更新部224は、Windows システムのスタートアップグル



ープ（Windows システムの起動時に実行されるアプリケーションを含むグループ）に登録されており、Windows システムの起動時に履歴ログ222の内容を履歴データベース226に反映することによって、履歴データベース226を更新する。履歴データベース226に登録される内容は、パス名を含むファイル名、アクセス回数、アクセス日時、ユーザ名等である。

【0098】ユーザは、ファインダ部202やクリーナ部204のウィンドウからファイルのアクセス履歴を要求して、各ファイルのアクセス履歴を図5のように表示させることができる。従って、アクセス履歴に応じ各ファイルの要否を決定することができる。

【0099】H. クリーナ部204の詳細とファイル削除の手順：図9は、クリーナ部204の内部構成を示すブロック図である。クリーナ部204は、ユーザインタフェイス部230と、情報取得部232と、解析部234と、アンインストール部236と、バックアップ部238と、参照リンク240とを有している。ユーザインタフェイス部230は、クリーナ部204を利用する際の種々の画面を作成して表示する機能と、ユーザからの指示を受け取る機能を有する。また、情報取得部232は、Windows システムのプログラムマネージャや各種の設定ファイル（INI ファイル）から必要な情報を取得する機能を有する。解析部234は、後述する各種の解析を実行する機能と、参照リンク240を作成する機能とを有する。アンインストール部236は、削除対象として指定されたファイルの削除を実行する削除実行手段としての機能を有する。また、バックアップ部238は、バックアップの対象として指定されたファイルを書庫ファイルとして保存するバックアップ実行手段としての機能を有する。バックアップ部238は、前述したアーカイバ部208とは別個にクリーナ部204内に設けられている。

【0100】図10は、クリーナ部204を用いて行なうアプリケーションプログラムのアンインストール処理の手順を示すフローチャートである。ユーザがクリーナ部204のアンインストール機能を選択すると、ステップS1において図11に示すようなアプリケーションの選択画面（選択ダイアログ）が表示される。このダイアログの左側には、Windows システムのプログラムマネージャに登録されているアイコンがアウトライン形式でリスト表示される。すなわち、第1レベルとしてグループ名が表示され、第2レベルとしてそのグループ内のアイコンとタイトルとが表示される。グループ名の右側には、内部に含まれているアイコン数が表示される。

【0101】選択ダイアログの右端には、アンインストール時の実行方法を指定するチェックボックス（「試行」、「バックアップ」、「確認する」と記されたもの）と、左側のリスト表示で選択されているグループまたはアイコン（反転表示されているもの）の登録内容

と、コマンド実行用のボタンとが表示される。

【0102】図11に示すように、アイコンの登録内容は、タイトルと、コマンドラインと、実行時のディレクトリと、アイコンのファイル名とを含んでいる。一般に、Windows アプリケーションは、インストールされるとプログラムマネージャにそのアプリケーションのグループを作成し、その内部にアプリケーション起動用のアイコンを登録する。このアイコン登録には、上記の4つの内容が含まれている。「タイトル」とは、アプリケーションの名称である。「コマンドライン」とは、アプリケーションを起動するためのコマンドである。コマンドラインには、実行ファイルのファイル名が含まれているが、必要に応じて他のファイル名やパラメータが含まれている場合がある。コマンドラインの例については後述する。「実行時のディレクトリ」は、アプリケーションを実行するときの初期ディレクトリである。プログラムマネージャは、アプリケーションを起動する前にこのディレクトリにカレントディレクトリを設定してから起動する。従って、アプリケーションで読み出すファイルやアプリケーションで作成したファイルは、このディレクトリ内のファイルとして指定される。「アイコン」には、アイコンのファイル名とその中のアイコンの順番が保存されている。

【0103】図10のステップS2において、ユーザが選択ダイアログの「アンインストール」または「解析」のボタンを選択すると、カーソル位置によって選択されているグループまたはアイコンに登録されているアプリケーションの解析がステップS3で実行される。以下では、1つのアイコンを解析対象として選択した場合について説明する。グループが選択されている場合には、そのグループ内のすべてのアイコンに関して処理が行なわれる。

【0104】なお、図11に示す選択ダイアログボックスとマウス73は、本発明におけるアイコン選択手段の機能を実現している。

【0105】図12は、実施例におけるディレクトリ構造を示す説明図である。＜＞で囲まれたものはディレクトリ名を示し、囲まれていないものはファイル名を示している。この例では、ディスクドライブ「A:」に、2つのディレクトリ＜DISK2＞と＜Windows＞があり、ディレクトリ＜Windows＞の中には＜SYSTEM＞というサブディレクトリが設けられている。以下の説明では、このディレクトリ＜DISK2＞に格納されているアプリケーションプログラムのアンインストールを行なう場合について主に説明する。

【0106】図13は、ステップS3の詳細手順を示すフローチャートである。ステップS11では、アイコンの登録内容から、削除対象のメインファイルの解析が実行される。ここでは、アイコンに登録されているコマンドライン（図11）が調べられる。



【0107】図14は、ステップS11の詳細手順を示すフローチャートである。また、図15は、コマンドラインの例を示す説明図である。

【0108】アイコンに登録されているコマンドラインは、アイコンをマウスでクリックした時に実行されるコマンド（命令）である。コマンドラインは、次に示すような2つの形式のいずれかであるのが普通である。

【0109】① <実行ファイル>

② <実行ファイル>+<データファイル>

【0110】上記①のように、1つの実行ファイルのみがコマンドラインに記述されている場合には、そのアイコンの登録の対象となっているのは、その実行ファイルのアプリケーションプログラムそのものである。また、上記②のように、実行ファイルの他にデータファイルが記述されている場合には、実行ファイルのアプリケーションではなく、データファイルがアイコンの登録の対象となっていることが多い。例えば、データベースプログラムのデータベースがアイコンとして登録されている場合などがこれに該当する。この場合には、データファイルを優先して削除対象とすることが望ましい。また、②の場合には、実行ファイルがWindows システムの標準的な実行ファイルであることがあるので、実行ファイルを削除することはできない場合が多い。さらに、コマンドラインには、図15の例のように、実行ファイル（EXEファイル）やデータファイルの他に、実行時の条件を示すパラメータ等が含まれていることがある。

【0111】このように、コマンドラインには実行ファイルを示す文字列の他に種々の文字列が含まれているので、図14に示すような手順でコマンドラインの解析を行ない、アイコンの登録対象となっている実体的なファイルを削除対象メインファイルとして特定する。解析部234によるこのような解析機能は、本発明における削除対象メインファイル特定手段の機能に相当する。

【0112】図14のステップS31では、コマンドラインが複数の文字列を含むか否かが判断される。コマンドラインが1つの文字列しか含まない場合は、その文字列は実行ファイルのファイル名であるのが普通である。この場合には、ステップS32においてその文字列のファイルが実在するか否かが判断される。文字列で示されるファイルが実在するか否かを判断する際には、そのファイルをWindows システムで決められているパスサーチ順を考慮して検索する。

【0113】Windows システムのパスサーチ順とは、Windows システム上でプログラムを実行する場合に、その実行ファイルのファイル名がフルパス（ディスクドライブ名からディレクトリまで含む完全なパス）で指定されていない場合に、その実行ファイルを探す順序を言う。具体的には、次の順序でパスがサーチされる。

【0114】（1）カレントディレクトリ。

（2）Windows ディレクトリ（図12の<Windows

>）。

（3）Windows システムディレクトリ（図12の<SYSTEM>）。

（4）現在のタスクを実行している実行ファイルのあるディレクトリ（実施例では、ディスク整理プログラムのあるディレクトリ）。

（5）PATH環境変数内にリストされているディレクトリ。

（6）ネットワークに割り当てられているディレクトリ。

【0115】その文字列のファイルが実在する場合にはそのファイルが削除対象メインファイルとして特定され（ステップS34）、実在しない場合には削除対象メインファイルなしとされる（ステップS33）。

【0116】一方、コマンドラインが複数の文字列を含む場合には、ステップS41以降の処理が実行される。ステップS41では、まず最初の文字列について、そのファイルが実在するか否かが判断される。この際にもWindows システムのパスサーチ順で探索が行なわれる。その文字列で示されるファイルが実在すれば、ステップS42において、そのファイルが実行ファイルであるか否かが判断される。実行ファイルでなければ、そのファイルが削除対象メインファイルとして特定される（ステップS44）。一方、そのファイルが実行ファイルであれば、それ以外のファイルがアイコンの登録対象となっている可能性があるので、ステップS43において次の文字列があるか否かを調べる。次の文字列があれば、ステップS41とステップS42とを再度実行する。こうして、ステップS41～S43を繰り返すことによって、複数の文字列の中で、実行ファイルでなく実在するファイルを探索する。そして、実行ファイルでなく実在するファイルがあれば、そのファイルを削除対象メインファイルとする（ステップS44）。一方、実行ファイルでなく実在するファイルが存在しなければ、コマンドラインの1番目にある実行ファイルを削除対象メインファイルとする（ステップS45）。

【0117】なお、各文字列で示されるファイルが実在するか否かを判断する場合に、「/」（スラッシュ）や「-」（マイナス）で始まる文字列は実在しないものと判断され、ファイルの探索はスキップされる。これは、「/」や「-」で始まる文字列は実行時のパラメータだからである。

【0118】図14の手順によれば、コマンドラインが実行ファイルを示す1つの文字列のみを含むような場合には、その実行ファイルが削除対象メインファイルとして特定される。一方、図15に示す3つのコマンドラインの例のように、コマンドラインが実行ファイル以外の文字列を含んでいる場合には、実行ファイルでなく、かつ、実在するファイルを示す文字列があれば、その実在するファイルが削除対象メインファイルとして特定され

る。

【0119】図15(A)に示す例では、コマンドラインに2つの文字列が含まれており、最初の文字列で示される実行ファイルは実在する。この時、2番目の文字列で示されるファイルが実在すれば、そのファイルが削除対象メインファイルとなり、実在しなければ1番目の文字列で示される実行ファイルが削除対象メインファイルとして特定される。

【0120】図15(B)で示される例では、コマンドラインに5つの文字列が含まれており、最初の文字列で示される実行ファイルは実在する。2番目と3番目の文字列は「/」(スラッシュ)で始まっており、実行時のパラメータなので実在しないものと判断される。従って、残りの4番目と5番目の文字列で示されるファイルが実在するか否かに応じて、図15(B)に示すように削除対象メインファイルが特定される。

【0121】図15(C)で示される例では、コマンドラインに4つの文字列が含まれており、最初の文字列で示される実行ファイルは実在する。2番目の文字列はディレクトリを示しているため、そのファイルは実在しない。また、3番目、4番目の文字列もパラメータを示しているため、そのファイルは実在しない。この場合には、1番目の文字列で示される実行ファイルが削除対象メインファイルとして特定される。

【0122】なお、図14の手順では、コマンドラインの左端から文字列を調べるようにしていたが、コマンドラインの右端から文字列を調べるようにしてもよい。また、削除対象メインファイルを1つと限定する必要はなく、コマンドラインから複数の削除対象メインファイルを特定するようにしてもよい。例えば、コマンドラインに含まれるすべてのファイルを削除対象メインファイルとしてもよい。

【0123】こうして削除対象メインファイルが特定されると、図13のステップS12において、削除対象メインファイルが実行ファイルであるか否かが判断される。実行ファイルであればステップS13～S19の解析が実行され、実行ファイルでなければこれらのステップを省略してステップS20に移行する。

【0124】なお、ステップS12～S22における種々の解析処理の機能は、本発明における検索手段に相当する。

【0125】ステップS13では、参照DLL ファイルと参照VBX ファイルの解析が実行される。ここで、「VBX ファイル」とは、Visual Basic (マイクロソフト社の商標)用のカスタムコントロール(リストボックスやボタンなどプログラム作成の部品として利用できるプログラム)である。その実体部分は、Visual Basic本体と通信するためのAPI 関数(Application Interface 関数)によって呼び出されるDLL である。一般に、VBX はVisual Basicのプログラム内で利用が宣言され、Visual Basic

で作成した実行ファイルに参照情報が書き込まれている。しかし、この形式は、実行ファイルからのDLL への参照とは異なる形式で書き込まれている。このため、アプリケーションプログラムによるVBX の参照関係は、DLL とは別に調査する。

【0126】図16は、ステップS13の詳細手順を示すフローチャートである。ステップS51～S53では、実行ファイル内に含まれているインポート名テーブルとリロケーションテーブルとを調べて、DLL やVBX を参照しているか否かを判断する。

【0127】図17は、インポート名テーブルとリロケーションテーブルの構成を示す説明図である。実行ファイルやDLL ファイルは、MS-DOSヘッダと、情報ブロックと、セグメントテーブルと、モジュール参照テーブルと、インポート名テーブルと、コードセグメントと、リロケーションテーブル等の種々のブロックが含まれている。情報ブロックには、これらの各ブロックの位置を示す情報が登録されている。

【0128】インポート名テーブルには、DLL の名前や関数名などの名前が可変長で登録されており、また、その名前の長さが登録されている。モジュール参照テーブルには、各DLL や各関数について、インポート名テーブルの先頭アドレスからのオフセットが登録されている。これは、インポート名テーブル内の名前が可変長だからである。

【0129】リロケーションテーブルは、コードセグメントの後に設けられている。リロケーションテーブルには、項目数と、アドレス、リロケーションタイプ、コードセグメント内の参照位置、モジュール参照テーブルのインデックス、および、インポート名テーブルのオフセットまたはDLL 内の関数の順序数が登録されている。アドレスタイプとリロケーションタイプは、実行ファイルがDLL や関数を参照する方法を示す情報である。コードセグメント内の参照位置は、コードセグメント内においてDLL や関数を参照している位置を示している。モジュール参照テーブルのインデックスは、そのDLL や関数の情報が、モジュール参照テーブルの何番目の情報として登録されているかを示している。インポート名テーブルのオフセットは、モジュール参照テーブルに登録されているものと同じ情報である。これは、モジュール参照テーブルを使用せずに、直接インポート名テーブルを参照する場合に使用される。DLL 内の関数の順序数とは、DLL に含まれている複数の関数の1つを使用する場合に、その何番目の関数を使用するかを示す数である。

【0130】参照DLL を調べる場合には、リロケーションテーブル内に登録されているモジュール参照テーブルのインデックスから、モジュール参照テーブル内に登録されているオフセット位置を調べる。そして、インポート名テーブルから、そのオフセット位置に登録されている名前を読み出すことによって、参照しているDLL や関

数を調べることができる。

【0131】図16のステップS51では、上述の方法によってインポート名テーブルの中を調べ、そこに含まれているDLLのファイル名を解析部234内部の参照DLLのリストに加える。ステップS52では、インポート名テーブル内にVBRJP200.DLLやVBRUN200.DLLなどのVisual BasicのDLLの情報が含まれているかどうかを調べる。Visual BasicのDLLの情報が含まれていれば、VBXを参照している可能性があるので、後述するステップS54における実行ファイル内の解析を行なう。

【0132】ステップS53では、リロケーションテーブルの中で、LoadLibrary()というAPI関数を利用しているかどうかを調べる。Windowsシステムでは、LoadLibrary()を利用することで、実行時に動的にDLLを指定してその内部の関数を呼び出すことができる。従って、このLoadLibrary()がリロケーションテーブルに含まれていれば、ステップS54における実行ファイル内の解析を行なう。

【0133】ステップS54では、実行ファイルの内部を解析して「.DLL」、「.VBX」の文字列を検索する。これらの文字列が含まれている場合は、そのDLLファイルやVBXファイルが実際に存在するか否かをチェックした上で、参照DLLと参照VBXのリストに加える。参照DLLや参照VBXの存在をチェックする場合にも、前述したパスサーチ順でそのファイル名のファイルが探索される。このため、別々のディレクトリに同じ名称のDLLがあっても、互いに区別されることになる。

【0134】ところで、DLLファイルやVBXファイル自体が、他のDLLを参照している場合がある。そこで、ステップS55では、参照DLLと参照VBXのそれぞれについて、ステップS51～S54と同様の処理を行なって、参照DLLや参照VBXに含まれているDLLやVBXを参照リストに加える。そして、この処理を未チェックのDLLやVBXがなくなるまで実行する。

【0135】参照DLLの中には、Windowsシステムが提供するDLLファイルも含まれている(GDI.EXE、COMMDLG.DLLなど)。これらのDLLファイルは削除してはならないので、解析部234はこれらのファイル名をあらかじめ記憶しており、参照DLLの解析対象から除外する。

【0136】ここで、図12に示すディレクトリ構造を対象にして、参照DLLと参照VBXとを解析する例を説明する。削除対象メインファイルがディレクトリ<DISK2>に含まれている実行ファイル「DXUTW.EXE」であるものと仮定する。まず、ステップS51でこの実行ファイルDXUTW.EXEのインポート名テーブルを調べると、VBRJP200.DLLが含まれていることが解るので、これを参照DLLのリストに加える。VBRJP200.DLLはVisual BasicのDLLなので、ステップS54において実行ファイルDXUTW.EXEの内部の解析を行う。ここで、実行ファイルDXUTW.EXEの内部に、DXUTW.DLL、MSGBLAST.VBX、SPIN.VBX

、THREED.VBXの各文字列が見つかったと仮定する。これらの各文字列のファイルを上述したパスサーチ順で検索すると実在するので、これらを参照リストに加える。この時点での参照リストは図18に示すものになる。

【0137】以上の処理で、実行ファイルDXUTW.EXEが直接参照しているDLLとVBXが判明した。次に、ステップS54において、見つかった各参照ファイルについて、それぞれがさらに参照しているDLLとVBXを探索する。VBRJP200.DLLのインポート名テーブルには、KERNEL、GDI、USER、KEYBOARD、WIN87EMなどのDLLが含まれているが、これらはWindowsシステムが提供するDLLファイルなので、参照リストには追加されない。また、VBRJP200.DLLのリロケーションテーブルを調べると、LoadLibrary()が使用されていることがわかるので、VBRJP200.DLLの内部サーチを行うと、VBRJP200.DLLが見つかる。しかし、これは自分自身なので参照リストには追加されない。このようにして、図18に示す各ファイルについて参照DLLと参照VBXとが調べられる。この結果、実行ファイルDXUTW.EXEが参照しているDLL、VBXの参照リストは、最終的に図18に示したリストと同じものとなる。

【0138】こうして参照DLLと参照VBXの解析が終了すると、図13のステップS14においてINIファイルの解析が行なわれる。アプリケーション専用のINIファイルは、Windowsディレクトリ内、または、アプリケーションの実行ファイルと同じディレクトリ内に、アプリケーションの実行ファイル名と同じ名前で作成されるのが普通である。例えば、実行ファイル名が「APP1.EXE」であれば、通常は「APP1.INI」というINIファイルが作成される。そこで、ステップS14では、実行ファイルのファイル名の拡張子部分を「INI」に変更した名称を有するファイルを、①Windowsディレクトリ、②実行ファイルのあるディレクトリ、の順に探し、そのファイルを見つけた場合に削除候補とする。

【0139】なお、INIファイルのファイル名に実行ファイルのファイル名とは異なる名前が付けられているアプリケーションプログラムも若干存在する。そのようなアプリケーションプログラムに関しては、解析部234がそのアプリケーションのINIファイル名を記憶しておき、その実行ファイルが削除対象メインファイルとして特定された場合に、そのINIファイル名を削除候補とする。

【0140】削除対象メインファイルが図12に示す実行ファイルDXUTW.EXEである場合には、このファイル名の拡張子部分を「INI」に変更したINIファイルは「DXUTW.INI」となる。このファイルを図12のディレクトリ構造から探すと、Windowsディレクトリに見つかるので、そのファイルを削除候補とする。

【0141】図13のステップS15では、ヘルプファイルの解析が行なわれる。ヘルプファイルは、一般的

に、アプリケーションプログラムの実行ファイル名の拡張子を「HLP」に変えたものが利用されるので、その名称のファイルが実行ファイルのあるディレクトリ内に存在するか否かを調べ、存在すれば削除候補とする。削除対象メインファイルが実行ファイルDXUTW.EXEである場合には、このファイル名の拡張子部分を「HLP」に変更したファイルは「DXUTW.HLP」となる。このファイルを図12のディレクトリ構造から探すと、ディレクトリ<DISK2>内に見つかるので、このファイルを削除候補とする。

【0142】図13のステップS16では、アプリケーションプログラムの実行ファイルとデータファイルの関連付けの解析が行なわれる。前述した図8に示すように、この関連付けの情報は、Windows システムのINI ファイルであるWIN.INI内の[Extensions] セクションに次の形式で登録されている。

【0143】<拡張子>;=<実行ファイル名>; ^.<拡張子>;

【0144】そこで、WIN.INI内の[Extensions] セクションを解析して、削除対象メインファイルを<実行ファイル名>;とするエントリ(記述)が存在かどうかを調べる。そのエントリが存在する場合には、そのエントリを削除候補とする。例えば、削除対象メインファイルが実行ファイルDXUTW.EXEである場合を考えると、図8に示すように、この実行ファイルに対応するエントリ「dxt=a:¥DISK2¥DXUTW.EXE ^.dxt」が含まれているので、このエントリを削除候補とする。

【0145】図13のステップS17では、拡張子のファイルの解析が行なわれる。特定の拡張子に関連付けられた実行ファイルを削除すると、その拡張子のファイル自体が取り扱えないようになる。このため、削除対象メインファイルに対する関連付けをWIN.INIの[Extensions]セクションから削除する場合には、その拡張子を有するファイルも削除候補とする。ただし、ステップS17ではファイルが実際に存在するかどうかは調べず、単に「拡張子<xxx>;のファイル」が削除候補になる旨を登録する。

【0146】図13のステップS18では、WIN.INI ファイル内のセクションの解析が行なわれる。アプリケーションプログラムによっては、個別のINI ファイルを使用せずに、WIN.INI 内にその環境設定の内容を書き込むものがある。その場合には、アプリケーションプログラムの実行ファイル名から拡張子部分を除いたものをセクション名とする場合が多いので、WIN.INI 内にそのようなセクションが存在するかどうかを調べ、存在する場合にはそのセクションを削除候補とする。例えば、削除対象メインファイルが実行ファイルDXUTW.EXEである場合を考えると、図8に示すように、この実行ファイル名に対応するセクション[DXUTW]がWIN.INI内に存在する。従って、このセクションが削除候補として解析部234内に登録される。

【0147】図13のステップS19では、アプリケーションディレクトリ内のファイルの解析が行なわれる。アプリケーションプログラムの実行ファイルを削除対象メインファイルとする場合には、その実行ファイル以外にも不必要になるファイルが存在する可能性がある。このため、その実行ファイルを含むディレクトリがWindows ディレクトリ(図12の<Windows>;)やWindows システムディレクトリ(図12の<SYSTEM>;)でない場合には、そのディレクトリ内のファイルを削除候補として登録する。例えば、削除対象メインファイルが図12に示す実行ファイル「A:¥DISK2¥DXUTW.EXE」である場合には、そのアプリケーションディレクトリは、A:¥DISK2である。そこで、解析部234は、このディレクトリを以下のような形で削除候補のリストに加える。

【0148】「ディレクトリ内のファイル A:¥DISK2 (個別に選択する)」

【0149】後述する削除候補の確認処理(図10のステップS5)では、このディレクトリ内のファイルの中で、他の削除候補には含まれないファイルが一覧表示され、そこからユーザーが削除するファイルを選択する。図12に示すディレクトリ「A:¥DISK2」内のファイルの中で、DXUTW.DLL は参照DLL ファイルとして削除候補となっており、DXUTW.HLP はヘルプファイルとして削除候補となっている。従って、ディレクトリ「A:¥DISK2」内のその他の削除候補ファイルとしては、「XDRVOTPW.EXE」、「XDRVOPT.DLL」、「README.WRI」の3つが一覧表示されることになる。

【0150】図13のステップS20では、OLE 情報の解析が行なわれる。ここで、OLE とは、「オブジェクトとリンクの埋め込み」と呼ばれるWindows システムの機能である。ここで、「オブジェクト」とはアプリケーションプログラムで作成された絵やスプレッドシートなどの1つの単位の情報を意味する。OLE 機能を使用するアプリケーションプログラムには、オブジェクトを別の文書に埋め込んだりリンクしたりすることのできる「サーバー」と、他のアプリケーションが作成したオブジェクトを受け入れることのできる「クライアント」とがある。アプリケーションプログラムによってはサーバーにもクライアントにもなれるものがあり、また、いずれか一方にしかなれないものもある。OLE 機能のサーバーになれるアプリケーションプログラムは、インストール時に、Windows システムの登録データベース「REG.DAT」(「レジストリファイル」とも呼ぶ)にその登録を行う。

【0151】登録データベースREG.DAT には、Windows システムのための各種の情報が登録されているが、その中に各アプリケーションプログラムのOLE 機能に関する登録も含まれている。実行ファイルを削除する場合には、登録データベースREG.DATにおけるOLE 情報の登録が無意味になるので、この登録情報も削除候補とする。

具体的には、登録データベースREG.DATの登録内容を調べて、OLE機能で起動するアプリケーション名が削除対象メインファイルとなっている情報を削除候補とする。例えば、実行ファイルDXUTW.EXEが削除対象メインファイルである場合に、この実行ファイルに関して図19に示すようなOLE情報が登録データベースREG.DATに登録されているものとする。この時、図19に示す3つの登録内容が削除候補として解析部234内に登録される。図19のOLE情報の第1行目の記述は、『拡張子「.dxt」のファイルは「Disk2.Document」というクラスのドキュメントファイルである』という意味を有している。また、第2行目は、『Disk2.Documentクラスのドキュメント名は「Disk2 ドキュメント」である』という意味を有しており、第3行目は『Disk2.DocumentクラスのファイルはアプリケーションA:¥DISK2¥DXUTW.EXEにより編集できる』という意味を有している。

【0152】図13のステップS21では、図10のステップS2で選択されたアイコンの解析が行なわれる。図11に示したように、ステップS2においてプログラムマネージャのグループやアイコンを選択してアンインストールを指定した場合には、そのアイコンに関するプログラムマネージャ内の登録内容が削除候補のリストに加えられる。アイコンに関する登録内容は、図11に示す、「タイトル」、「コマンドライン」、「実行時のディレクトリ」、「アイコン」等の項目の内容である。また、「アイコン」で示されるアイコンファイルの拡張子が「.ico」である場合には、そのアイコンファイルはアイコンの画像を表すグラフィックファイルである。従って、そのアイコンファイルも削除候補に加える。図11の例では、アイコン登録情報のタイトルは「Disk ユーティリティ」なので、このタイトルのアイコンに関するプログラムマネージャの登録内容を削除候補に入れる。また、図11では、アイコンファイルが「DXUTW.EXE」として指定されているが、これは拡張子が「.ico」ではないので、削除候補とはしない。拡張子が「.ico」ではないファイルを削除候補としないのは、そのファイルがアイコン画像の作成以外の目的で使用されている可能性が高いからである。

【0153】図13のステップS22では、アプリケーションディレクトリの解析が行なわれる。ファイルの削除によってディレクトリが空になる場合は、そのディレクトリも削除できる。従って、削除対象メインファイルを含むディレクトリがWindowsディレクトリやWindowsシステムディレクトリでない場合には、そのディレクトリ自体が削除候補として追加される。例えば、削除対象メインファイルが図12に示す「A:¥DISK2¥DXUTW.EXE」である場合には、そのディレクトリ名「A:¥DISK2」が以下のような形で削除候補リストに加えられる。

【0154】「ディレクトリ A:¥DISK2 (ディレクトリが空になったら削除)」

【0155】なお、図13の手順からも解るように、ステップS13～S20における解析は削除対象メインファイルが実行ファイルである場合にのみに行なわれるが、ステップS21、S22の解析は、削除対象メインファイルが実行ファイル以外の場合にも実行される。

【0156】図20は、以上の解析の結果として得られた削除候補の内容を示す説明図である。なお、図20(A)～(D)、(F)、(H)の削除候補はファイルであり、図20(K)の削除候補はディレクトリ、図20(E)、(G)、(I)、(J)の削除候補は所定のファイル(WIN.INI、REG.DATAおよびプログラムマネージャ)内に登録されている情報である。このように、実施例の解析部234は、個々のファイルやディレクトリを削除候補とするばかりでなく、削除対象メインファイルに関して所定のファイル内に登録されている情報も削除候補として特定するので、ユーザは、アプリケーションのアンインストール処理において、不要なファイルや情報をより効率的に削除することができる。

【0157】こうして、図10のステップS3における解析が終了すると、ステップS4においてデフォルト削除候補が決定される。ステップS3における解析の結果としては、図20に示したような削除候補が解析部234内に登録されているが、この削除候補の中にも削除すべきでないファイルや情報が存在する可能性がある。そのため、ステップS4では、解析で得られた各削除候補ごとに、削除候補として実際に表示するかどうかを解析部234が判定する。

【0158】図20(A)に示す削除対象メインファイルは、常にデフォルトの削除候補とする。図20(B)に示す参照DLLと参照VBXについては、まず、参照リンク240(図9)に登録があるかどうかを調べる。

【0159】参照リンク240は、実行ファイルが参照するDLLファイルやVBXファイルについての情報を登録したデータベースである。図21は、参照リンク240の登録内容を示す説明図である。参照リンク240には、図21(A)に示す実行ファイルリストと、図21(B)に示す被参照ファイルリストが含まれている。実行ファイルリストには、参照関係の調査の対象となった実行ファイルが登録されている。また、被参照ファイルリストには、実行ファイルリストに登録されている実行ファイルから参照されているDLLファイルおよびVBXファイルと、参照されている回数とが登録されている。なお、参照リンク240の作成方法については、後述する。

【0160】削除対象メインファイルが参照リンク240の実行ファイルリストに含まれてる場合には、削除候補となっている参照DLLや参照VBX(図20(B))の被参照数を調べる。被参照数が1であれば、1つの実行ファイルから参照されているだけなので、そのDLLやVB

X をデフォルトの削除候補とする。一方、被参照数が2以上の場合には、削除できないので、デフォルトの削除候補から除外する。

【0161】削除対象メインファイルが参照リンク240の実行ファイルリストに含まれていない場合には、参照リンク240を使用することはできない。この場合には、ステップS3における解析で得られた参照DLL や参照VBX を他の実行ファイルが使用している可能性があるため、参照DLL や参照VBX はデフォルトの削除候補から除外される。また、削除対象メインファイルが参照リンク240の実行ファイルリストに含まれていても、削除候補となっている参照DLL や参照VBX が被参照ファイルリストに含まれていない場合にも、他の実行ファイルが使用している可能性があるため、これらをデフォルトの削除候補から除外する。

【0162】なお、図10のステップS2において複数の実行ファイルを含むグループが指定された場合には、図13のステップS11において複数の実行ファイルが削除対象メインファイルとして特定される。このときには、これらの実行ファイルからの参照数の合計値と、参照リンク240に登録されている被参照数とが一致しているか否かを調べる。例えば、図12に示すディレクトリ<DISK2>を削除対象として指定した場合には2つの実行ファイル「DXUTW.EXE」、「XDRVOTPW.EXE」が削除対象メインファイルとなる。この時、2つの実行ファイルがどちらも「DXUTW.DLL」を参照している場合には、「DXUTW.DLL」の被参照数は2となる。従って、参照リンク240に登録されている被参照ファイル「DXUTW.DLL」の被参照数が2であれば、この被参照ファイルをデフォルトの削除候補とする。

【0163】なお、図20(C)～(K)に示すファイルと情報は、すべてデフォルトの削除候補とされる。

【0164】デフォルトの削除候補が決定されると、その削除候補とそれ以外のファイルおよび情報が区別して表示される。図22は、削除候補を示す画面の一例を示す説明図である。なお、図22では図示の便宜上、図20に示す削除候補とは異なるファイル名が表示されているが、実際には、図20に示す削除候補が画面に表示されるはずである。

【0165】画面の上部には、削除対象メインファイルのファイル名「A:\DISK2\DXUTW.EXE」とそのタイトルが表示されている。また、画面の中央部には、削除対象メインファイルの実行に伴って使用されるすべてのファイルや情報について、カテゴリと削除内容が表示されている。カテゴリとは、DLL ファイル、INI ファイル、ヘルプファイル、アイコンファイル等のファイルの種類である。削除内容の欄には、パス名を伴ったファイル名と注意書きとが書き込まれている。注意書きには、現在実行中で削除対象として選択できない旨や、他のプログラムで使用されているので削除対象として選択できない旨な

どが記載されている。デフォルトの削除候補とされたファイルや情報の行は反転表示されている。なお、図22(A)の画面と図22(B)の画面では、削除候補一覧の中の異なる部分が表示されている。

【0166】ユーザは、各ファイルの行をマウス73でクリックすることによって、反転表示されている削除候補を削除対象から除外したり、削除候補でないものを削除対象として追加したりすることができる。なお、図22(B)のカテゴリに「ディレクトリ内のファイル」と書かれているディレクトリ「A:\DISK2」を選択すると、その行の下に、そのディレクトリ「A:\DISK2」に含まれているファイルの中ですでに表示されているファイルを除くすべてのファイルが一覧表示される。従って、ユーザは、その一覧表示の中から削除対象とするファイルを選択することができる。

【0167】なお、図22の画面表示を用いてマウス73で削除対象を選択する機能は、本発明における削除対象選択手段に相当する。

【0168】図22の画面上で削除対象を決定した後、ユーザが画面右端の「アンインストール」ボタンをマウス73で押すと、図10のステップS5～S9の処理が実行される。なお、ステップS5～S9の中のどれが実行されるかは、図22の画面の上方にある3つのチェックボックス「試行」、「確認する」、「バックアップ」のチェック状態によって異なる。チェック状態に応じた処理の手順についてはさらに後述するが、以下ではまず、ステップS5～S9がすべて実行される場合について説明する。

【0169】ステップS5では、削除対象として選択された各ファイルや情報について、図23に示すような確認ウィンドウが表示される。ユーザは、各削除対象について、削除するか否かを個別に確認することができる。また、そのファイルの内容を再度確認したい場合には、図23の「詳細情報」ボタンをマウス73を用いて押すことによって、前述した図7に示すようなファイルの詳細情報を見ることができる。さらに、そのファイルが、ワードプロセッサ、グラフィックス、表計算プログラム等のアプリケーションプログラムで作成されたデータファイルである場合には、図7の「表示」のボタンを押すことによって、アプリケーションプログラムを立上げること無くその内容(文書、画像、スプレッドシート等)を画面上に表示することができる。

【0170】こうして削除対象となっている各ファイルや情報の確認処理が終了すると、図10のステップS6において、クリーン情報ファイルが作成される。アンインストール処理では、ファイルの削除だけでなく、各種の設定ファイル(INI ファイル)や登録データベースREG.DAT の修正も行なわれる。そこで、アンインストール処理の復元を可能とするために、ファイルに対して行った修正等の内容を示すクリーン情報ファイルが作成され

る。このクリーン情報ファイルは、ステップS7におけるバックアップ時に、他のファイルと共に保存される。そして、アンインストール処理を復元する時には、バックアップファイルを復元した後で、このクリーン情報ファイル内容を解析することによって、設定ファイルや登録データベースの内容を元の設定状態に戻すことができる。

【0171】図24は、クリーン情報ファイルに含まれる内容を示す説明図であり、図20に示す削除候補がすべて削除対象として確認された場合の例である。図24(A), (B), (C), (D)は、それぞれ図20(G), (E), (I), (J)に対応する内容である。また、図24(E)の「プログラムマネージャのグループ削除」は、図10のステップS2においてグループ全体を削除対象として選択した場合に、そのグループ名が保存されることを意味している。

【0172】クリーン情報ファイルに含まれる内容は、アプリケーションプログラムの使用環境をアンインストール処理直前の環境に設定するための情報であり、アプリケーションプログラムの再インストールによっては復元されない可能性のある情報である。図24に示す各種の設定内容や登録内容は、アプリケーションプログラムのインストール後にユーザによって変更されている可能性がある。従って、図24に示す内容を保存しておけば、アプリケーションプログラムを再インストールした後に、これらの設定内容を復元することによって、アプリケーションプログラムをアンインストール処理直前の使用環境に復元することができる。

【0173】なお、アンインストール部236によるステップS6の処理機能は、本発明におけるクリーン情報ファイル作成手段に相当する。

【0174】図10のステップS7では、削除対象ファイルのバックアップが実行される。この時点では、最終的に削除するファイルや情報が確定している。そこで、ステップS7ではまず、バックアップの条件を設定するためのバックアップダイアログ（バックアップ条件設定ウィンドウ）が表示され、ユーザはここでバックアップ先と、バックアップレベルの入力を行なう。バックアップ先は、バックアップファイルを格納するためのディスクドライブ名である。削除される各ファイルに対するバックアップファイルのデフォルトのファイル名は、元のファイル名とは無関係に「DXmmdd??」に設定される。ここで、先頭の2文字「DX」はこのディスク内容整理システムによるバックアップファイルであることを示す接頭詞、次の4文字「mmdd」はファイルを削除する月と日を示す各2字の数字、「??」は同時に削除される複数のファイルに割り当てられる00から始まる通し番号、「ZIP」はZIP形式の書庫ファイルであることを示す拡張子である。ユーザは、必要であれば、この名前を変更することも可能である。

【0175】バックアップダイアログでは、さらに、バックアップのレベルを指定することができる。バックアップレベルとしては、フルバックアップと簡易バックアップがある。

【0176】フルバックアップでは、削除するファイルおよび修正するファイル（ファイル内の情報を削除・変更するファイル）をすべてバックアップ対象とする。

【0177】簡易バックアップでは、アプリケーションプログラムを再インストールすれば復元できるファイルについてはバックアップせず、ユーザが変更したと推定されるファイルのみバックアップを行う。したがって、通常ユーザが変更しないファイルはバックアップの対象から除外される。この結果、バックアップの対象となるファイルのサイズをかなり小さくすることができる。

【0178】簡易バックアップにおいて常にバックアップの対象としないファイルは、実行ファイル、ヘルプファイル、DLL ファイル、VBX ファイルである。これらのファイルの内容はユーザが変更できないので、アプリケーションプログラムを再インストールすれば同じファイルを復元できるからである。

【0179】簡易バックアップにおいて常にバックアップの対象とするファイルは、INI ファイル、および、そのアプリケーションプログラムに関連付けられた拡張子を有するファイルである。INI ファイルの設定内容はユーザによって変更される可能性があり、また、アプリケーションの拡張子を有するファイルは、アプリケーションのインストール後に作成された可能性が高いからである。

【0180】簡易バックアップにおいてバックアップの対象とすべきか否かが一見して不明なファイルもある。これには、アイコンファイルと、アプリケーションディレクトリ内の残りのファイルが含まれる。これらのファイルについては、次の判定条件に従ってバックアップの対象とするか否かを決定する。

【0181】[バックアップ要否の判定条件]: アプリケーションのディレクトリの更新日時よりも更新日時が新しいファイルは、バックアップ対象とする。それ以外のファイルは、バックアップ対象としない。

【0182】アプリケーションのディレクトリの更新日時は、アプリケーションのインストールを行った日時に一致するのが普通なので、上記の判定条件を使用すれば、インストール後に変更されたファイルのみをバックアップの対象とすることができる。

【0183】なお、削除対象とされたすべてのファイルに関して上記の判定条件を適用することによって、簡易バックアップの対象を決定するようにしてもよい。この方法によっても、実質的に上述の方法と同じ結果が得られる。

【0184】なお、バックアップ部238によって上述



のように簡易バックアップの対象を特定する機能は、本発明における簡易バックアップ特定手段に相当する。

【0185】バックアップ部238(図9)は、バックアップダイアログにおけるユーザの指定に基づいてバックアップ処理を実行する。具体的には、バックアップの対象となる全ファイルについて、ZIP形式の書庫ファイルを作成する。なお、バックアップ部238で作成された書庫ファイルの情報は、クリーナ部204からアーカイバ部208(図2)に転送され、アーカイバ部208によってその情報が管理される。

【0186】図10のステップS8では、アンインストール部236(図9)がアンインストール処理を実行する。すなわち、削除対象として確認されたファイルの削除、および、所定のファイル内の各種の情報の削除や修正等を実行する。

【0187】ステップS9では、解析部234によって参照リンク240(図9)の更新が実行される。前述した図21に示すように、参照リンク240には、実行ファイルリストと、各実行ファイルから参照されるDLLファイルおよびVBXファイルのリストが含まれている。ステップS8におけるアンインストール処理において実行ファイルが削除された場合には、実行ファイルリストからその実行ファイルを削除し、また、その実行ファイルが参照しているファイルの被参照数をそれぞれ1つ減算する。このようにして参照リンク240を更新しておけば、次の実行ファイルをアンインストールする際に、この参照リンク240を使用してDLLファイルやVBXファイルの削除の可否を決定することができる。

【0188】こうして、図10に示すアンインストール処理の工程がすべて終了すると、統計何バイトの領域が開放されたかが画面に表示されて、アプリケーション選択画面(図11)に戻る。

【0189】ところで、図10のステップS5～S9の工程は、図22の画面の上方にある3つのチェックボックス「試行」、「確認する」、「バックアップ」のチェック状態によって、以下のように選択的に実行される。

【0190】チェックボックスの「試行」がチェックされている場合には、ステップS6(クリーン情報ファイルの作成処理)、ステップS7(バックアップ処理)、ステップS8(アンインストール処理)、および、ステップS9(参照リンクの更新)を実行しない。従って、この「試行」のボックスをチェックしておけば、実際にファイルの削除を行わずに、どのようなファイルの削除や修正(ファイル内の情報の削除に伴う修正)が行なわれるかを知ることができる。

【0191】一方、「試行」がチェックされていないければ、ステップS6～S9が実行される。

【0192】チェックボックスの「バックアップ」がチェックされており、かつ、「試行」がチェックされていないければ、ステップS7のバックアップ処理が実行され

る。「バックアップ」がチェックされていないければ、ステップS7は実行されない。

【0193】チェックボックスの「確認する」がチェックされている場合には、ステップS5(確認処理)が実行され、チェックされていないければステップS5が実行されない。

【0194】このように、この実施例におけるディスク内容整理システムでは、ユーザが、バックアップの要否を指定したり、アンインストールを実際に行わずに削除されるファイルや情報のリストのみを作成したりすることができるので、ユーザがハードディスク84の内容を容易に整理することができる。

【0195】I. 参照リンクの作成方法: 図25は、解析部234による参照リンク240(図9、図21)の作成手順を示すフローチャートである。参照リンク240は、アンインストール処理のたびに毎回作成する必要はないので、アンインストール処理とは別途作成される。具体的には、ユーザがメニューから参照リンク240の作成を指示した時に図25の処理が実行され、プログラムマネージャに登録されている全アイコンについて、図13に示すステップS13の参照DLLの解析と同様の処理が行なわれる。なお、以下の説明では、DLLファイルとVBXファイルを含める意味で「DLLファイル」という単語を使用する。

【0196】図25のステップS60では、プログラムマネージャから次のアイコンを処理対象として取り出す。未処理のアイコンがなくなったら図25の処理を終了する。ステップS61では、処理対象のアイコンのコマンドラインを解析して、そのメインファイルを求める。ステップS61の処理は、図14に示す削除対象メインファイルの解析と実質的に同じ処理である。メインファイルが実行ファイルでない場合にはステップS62からステップS60に戻り、実行ファイルである場合にはステップS63以降の処理が実行される。これは、参照リンク240の作成処理においては、実行ファイルとDLLファイルの参照関係のみを調べればよいからである。

【0197】ステップS63では、処理対象の実行ファイルが参照リンク240の実行ファイルリスト(図21)に登録されているか否かを調べる。既に実行ファイルリストに登録されている場合には、ステップS61に戻り、次のアイコンが処理対象として選択される。処理対象の実行ファイルが実行ファイルリストに登録されていない場合には、その実行ファイルを実行ファイルリストに追加する(ステップS64)。

【0198】ステップS65では、処理対象の実行ファイルを調べて、参照しているDLLファイルを1つ得る。この処理は、前述した図13のステップS13における解析処理と同様に、実行ファイル内に含まれているインポート名テーブルとリロケーションテーブルを調べるこ



とによって行なわれる。ステップS66では、そのDLLファイルが参照リンク240の被参照ファイルリスト(図21)に登録されているか否かを調べる。そのDLLファイルが被参照ファイルリストに登録されていないければ、それを登録して(ステップS67)、被参照数を1つ増やす(ステップS68)。一方、そのDLLファイルが被参照ファイルリストに登録されていれば、その被参照数を1つ増やす(ステップS68)。

【0199】なお、DLLファイル自体も他のDLLファイルを参照することがある。そこで、ステップS69では、ステップS65で得られたDLLファイルが他のDLLファイルを参照しているか否かを調べる。参照DLLがあれば、ステップS66～S69が再帰的に繰返される。

【0200】ステップS70では、処理対象の実行ファイルが参照している次のDLLが存在するか否かを調べ、存在すればステップS66に移行する。ステップS66～S70を繰返し実行して、1つの実行ファイルに関する処理が終了すると、ステップS61に戻り、他のアイコンについてステップS61～S70の処理が繰返される。こうして、プログラムマネージャに登録されているすべてのアイコンについての処理が終了すると、そのリストの内容を参照リンク240に書き出して終了する。

【0201】このようにして参照リンク240を作成しておけば、実行ファイルの削除に伴ってDLLファイルやVBXファイルを削除候補とするか否かを容易に判断することが可能である。

【0202】J. オペレーティングシステムによる違い: 上記実施例では、Windows3.1システムの下で動作するコンピュータシステムを対象として説明していたが、この発明はWindows95(マイクロソフト社の商標)等の他のウィンドウズシステムの下で動作するコンピュータシステムにも適用可能である。この発明に関連するWindows3.1システムとWindows95システムとの違いは、以下の通りである。

【0203】①Windows95では、オペレーティングシステムが各ファイルの最終アクセス日を管理している。従って、上述した履歴データベース226に各ファイルの最終アクセス日を記録しておく必要はなく、オペレーティングシステムから提供される最終アクセス日を利用するようにすることができる。

【0204】②Windows3.1における登録データベースREG.DATは、Windows95では「レジストリ」と呼ばれるファイルに相当する。Windows95のレジストリには、さらに、Windows3.1においてWIN.INIやSYSTEM.INIに記述されていたシステム設定に関する内容も格納されている。上記の実施例において、登録データベースREG.DATに関する処理と、WIN.INIおよびSYSTEM.INI内のシステム設定に関する処理は、Windows95システムの下ではレジストリに対して行なわれる。この明細書における「レジス

トリファイル」という用語は、Windows3.1における登録データベースREG.DATとWindows95におけるレジストリとを含む広義の用語として使用される。

【0205】レジストリファイルには、WIN.INIに登録されているものと同様な関連付け情報(拡張子と実行ファイルとの関連を示す情報)も登録されている。図26は、レジストリファイルに登録されている関連付け情報を示す説明図である。図26(A)に示すように、レジストリファイル内の関連付け情報には、拡張子とドキュメントタイプの対応関係と、ドキュメントタイプとドキュメント名の対応関係と、ドキュメントタイプとその実行ファイルの対応関係と、が登録されている。この情報によって、拡張子と実行ファイルとが関連付けられる。なお、「%1」は、起動時の引数の指定である。関連付け情報で規定されている拡張子を有するデータファイルのファイル名が指定されて起動される時には、図26(A)の「%1」の部分が指定されたファイル名に置き換えられる。図26(B)は、DXUTW.EXEという実行ファイルと、拡張子dxtとの関連付け情報の実例である。例えば、「aaa.dxt」というファイルを起動すると、「a:\MSK2\DXUTW.EXE aaa.dxt」というコマンドラインが実行される。

【0206】このように、レジストリファイル(登録データベースまたはレジストリ)には、関連付け情報が登録されているので、前述した図13のステップS16における関連付け情報の解析は、レジストリファイルに対しても行なわれる。また、図20(I)において、OLE情報の他にレジストリファイル内の関連付け情報も削除候補となる。関連付け情報を実際に削除する場合には、図24(C)において、レジストリファイル内の関連付け情報もクリーン情報ファイルに含まれることになる。

【0207】③Windows95では、グループがプログラムマネージャではなくスタートメニューに登録される。すなわち、アイコンもスタートメニューに登録される(実際には、ファイルシステム上の特定の領域に書き込まれている)。従って、図20(J)において、スタートメニュー内のアイコンの登録内容が削除候補となる。また、アイコンを実際に削除する場合には、図24(D)において、スタートファイル内のアイコンの登録内容がクリーン情報ファイルに含まれることになる。

【0208】K. 書庫化アンインストール処理: クリーナ部204には、前述の図9において説明した各機能の他に、アプリケーションを書庫化してから削除を行なう書庫化アンインストール部300としての機能が備えられている。図27は、書庫化アンインストール部300の機能を示す説明図である。書庫化アンインストール部300は、アプリケーションとそれに関連するファイルおよび情報とを解析して削除候補とする機能と、削除候補を圧縮して書庫ファイルを作成するようにアークファ

イル部208(図2)に指示する機能と、アプリケーションを削除(アンインストール)する機能と、書庫ファイルの解凍に使用する解凍用アイコン310を登録する機能(アイコン変更手段)と、を有している。また、後述するように、書庫ファイルを解凍するようにアークファイル部208に指示する機能も有している。

【0209】図28は、書庫化アンインストール処理の手順を示すフローチャートである。ステップS101では、書庫化アンインストール処理の対象とするアプリケーションを選択する。これは、前述した図10のステップS1、S2の処理とほぼ同じである。書庫化アンインストール処理を選択するために、前述した図11に示すアプリケーション選択画面に「書庫化アンインストール」のボタンを追加しておくことが好ましい。なお、処理対象としては、1つのアイコンを選択することもでき、あるいはグループ内の複数のアイコンを選択することも可能である。以下では主に1つのアイコンを処理対象として選択した場合について説明する。

【0210】処理対象となるアイコン(またはグループ)が選択されると、ステップS102において、そのアイコンに登録されているアプリケーションの解析が行われる。これは、図10のステップS3～S6、S9の処理とほぼ同じである。すなわち、ステップS102では、図10のステップS3(解析)、ステップS4(デフォルト削除候補の決定)、ステップS5(ユーザによる削除候補の確認)、ステップS6(クリーン情報ファイルの作成)、および、ステップS9(参照リンクの更新)が行われる。なお、図10のステップS7(バックアップの実行)と類似の処理は後述するステップS103で実行され、ステップS8(リストの実行)と類似の処理は後述するステップS106で実行される。

【0211】ステップS103では、処理対象となるアイコンに関連しているファイルおよび情報の中で、削除候補となっているすべてのファイルおよび情報と、クリーン情報ファイルとを圧縮して書庫ファイルを作成する。すなわち、図20に示す削除候補の内容と、図24に示すクリーン情報とが圧縮される。この際、デフォルト削除候補か否か(これは図10のステップS4で決定される)には関係なく、図10のステップS3の解析の結果として、削除候補とされたすべてのファイルと情報が、1つの書庫ファイルとしてまとめられて圧縮される。換言すれば、ステップS103は、図10のステップS7におけるバックアップ処理を、すべての削除候補およびクリーン情報ファイルに対して行なったことに相当する。なお、圧縮と解凍の実際の処理は、アークファイル部208(図2)が実行する。

【0212】ステップS103において、カテゴリが「アプリケーションディレクトリ内のファイル」や「拡張子ファイル」であるファイルは、アプリケーションに

よって作成されたデータ(ドキュメント)であるのが普通である。従って、これらのカテゴリのファイルに関しては、ステップS102で行なわれるユーザによる確認処理(図10のステップS5)において選択されたファイルのみを書庫化するようにしてもよい。

【0213】また、カテゴリによっては、ユーザが削除候補の選択状態の変更することを禁止するようにしてもよい。例えば、カテゴリ「アイコン」のものを非選択状態にすると、後述ステップS105においてそのアイコンが削除されなくなってしまう。こうなると、元のアイコンと、解凍用アイコンとが並存するという不具合が生じる。このような不具合を防止するためには、カテゴリ「アイコン」のファイルの選択状態をユーザが変更できないようにしておけばよい。

【0214】なお、1つのグループ全体を処理対象として選択した場合には、そのグループ内のすべてのファイルをデフォルトで選択状態(デフォルトの削除候補)とすることが好ましい。

【0215】ステップS104では、解凍用アイコンをグループ内に登録する。解凍用アイコン(図27参照)は、元のアイコンの特定の位置(例えば右下)に、解凍用アイコンであることを示す特定のマーク(文字や記号)を付加したものである。また、アイコンの他の部分を元のアイコンの図柄とほぼ同じに保っておく。こうすれば、ユーザが解凍用アイコンを見た時に、どのアプリケーションの解凍用アイコンであるかを、容易に判別できる。

【0216】なお、ステップS101において、1つのグループ全体が選択された場合には、そのグループ内の全てのアイコンが、解凍するためのアイコンに置き換えられる。

【0217】解凍用アイコンのコマンドラインには、解凍プログラムの実行ファイルがコマンドラインの実行ファイルとして登録され、その実行ファイルで開かれるドキュメントとして書庫ファイルが登録される。こうすることによって、解凍時に解凍用アイコンをユーザが選択して起動した時に解凍プログラムが起動されて、対応する書庫ファイルを解凍することができる。グループ内のすべてのアイコンが解凍用アイコンに置き換えられている場合には、その内のいずれかが指定されると、グループ全体が解凍されるとともに、指定されたアイコンが起動される。このアイコンの実行中に、同じグループの別のアイコンが起動された場合には、既に解凍済なので、あらためて解凍を行わずに、そのプログラムの起動のみを実行する。

【0218】ステップS105では、レジストリファイルに登録されている関連付け情報を変更する。これは次のような理由による。前述したように、削除対象のアプリケーションで作成されたドキュメントファイルは削除候補とならないことがある。すなわち、ドキュメントフ

ファイルは、書庫化ファイルに含まれず、また、削除（アンインストール）もされないことがある。従って、ドキュメントファイルを開くようにユーザが指定しても、そのアプリケーション自体が書庫化されている場合には、そのドキュメントファイルを開くことができないことになる。ところで、図26（A）、（B）に示したように、レジストリファイルには、ドキュメントファイルの拡張子と、アプリケーションの実行ファイルとの関連付け情報が登録されている。そこで、ステップS105では、ドキュメントファイルの拡張子を、解凍プログラムの実行ファイルと関連付けるように、関連付け情報を変更する。こうすれば、ユーザがドキュメントファイルを開くように指示（ダブルクリック）した際に、対応するアプリケーションプログラムを起動できるようになる。

【0219】図26（C）は、変更後の関連付け情報を示している。図26（C）の最初の2行は図26（B）と同じであり、図26（C）の最後の2行が図26（B）から書換えられている。但し、図26（C）の最後の2行は、実際には1行で記述されているので、以下ではこれを「3行目」と呼ぶ。図26（C）の3行目の「＝」の左辺は図26（B）と同じである。図26（C）の右辺の最初の部分「C:\DXMELT.EXE」は解凍プログラムの実行ファイルを示しており、次の部分「/MELT "ARK.ZIP"」は解凍される書庫ファイルのファイル名を示している。もちろん、このファイル名「ARK.ZIP」としては、書庫ファイル毎に異なる名前が割り当てられる。図26（C）の3行目の最後の部分「/ARC a:\\$DISK2\DXUTW.EXE %1」の「」で囲まれた部分は、解凍後にプログラムを起動するためのコマンドラインを示しており、圧縮前の図26（B）の3行目の右辺と同じ内容を示している。このように、圧縮後の関連付け情報には、拡張子「.dx2」と、その拡張子に関連付けられたプログラムの実行ファイルを含む書庫ファイル「ARK.ZIP」と、解凍された実行ファイルを起動するためのコマンドライン「a:\\$DISK2\DXUTW.EXE %1」とを含んでいる。

【0220】図26（C）のように関連付け情報が書き換えられている時に、「.dx2」という拡張子を持ったドキュメントファイルを起動することが指定されると、図26（C）の3行目で指定されている書庫ファイル「ARK.ZIP」が解凍され、その後、コマンドライン「a:\\$DISK2\DXUTW.EXE %1」に従って、関連付けられたプログラムが起動される。このように、図26（C）のように関連付け情報を書き換えるようにしておけば、圧縮されていないドキュメントファイルの起動が指定された時に、その実行ファイルを含む書庫ファイルを解凍して、その書庫ファイル内の実行ファイルを自動的に起動することができる。従って、ユーザは、プログラムが圧縮されていることを意識せずに、ドキュメントファイルを利用できるという利点がある。

【0221】なお、上記のように圧縮後の関連付け情報

をすべてオペレーティングシステム用のレジストリファイル内に格納する代わりに、その一部を他のファイルに格納するようにすることも可能である。

【0222】図26（D）は、レジストリファイル内に格納される関連付け情報の他の例を示している。図26（D）の最後の2行（これらも実際は1行なので「3行目」と呼ぶ）では、図26（C）の3行目の右辺における「/MELT "ARK.ZIP" /ARC "a:\\$DISK2\DXUTW.EXE %1"」が「<##> /ARC」で置き換えられている。すなわち、図26（D）の例では、書庫ファイルのファイル名を示す文字列「/MELT "ARK.ZIP"」と、コマンドラインを示す文字列「"a:\\$DISK2\DXUTW.EXE %1"」とが省略されている。図26（D）の3行目における文字列「<##>」は、書庫ファイル番号を示している。アークファイル部208は、すべての書庫ファイルを書庫ファイル番号で管理している。また、書庫ファイル番号を含む管理ファイルは、解凍プログラムの実行ファイルDXMELT.EXEを含むディレクトリ（図26（D）の例ではC:\DXT）内に格納されている。従って、図26（D）のように、レジストリファイル内の関連付け情報の中に書庫ファイル番号を記述しておけば、拡張子.dxtを有するドキュメントの起動が指定された時に、図26（D）の関連付け情報に従ってこの書庫ファイル番号を有する書庫ファイルを解凍することができる。書庫ファイル内には、元の関連付け情報（図26（B））が含まれているので、書庫ファイルの解凍後に、この元の関連付け情報に従ってコマンドラインを実行することによって、アプリケーションが起動される。このように、図26（D）の例では、圧縮後の関連付け情報の一部（書庫ファイル番号）をレジストリファイルに格納し、他の一部（アプリケーションの実行ファイルのコマンドライン）を書庫ファイル内に格納し、さらに他の一部（書庫ファイル番号と書庫ファイルとの対応）を解凍プログラムの実行ファイルを含むディレクトリ内の管理ファイルに格納している。このように、レジストリファイルには、圧縮後の関連付け情報の少なくとも一部が含まれていればよい。但し、レジストリファイル内には、拡張子を示す情報と、その拡張子に関連付けられたアプリケーションを含む書庫ファイルを示す情報とを少なくとも含むようにすることが好ましい。

【0223】圧縮後の関連付け情報の一部（例えばコマンドライン）を、書庫ファイル内のクリーン情報ファイルに格納するようにしてもよい。この場合にも、レジストリファイル内の関連付け情報に、解凍すべき書庫ファイルを示す情報を含める。こうすれば、この情報に従って書庫ファイルを解凍するとともに、その書庫ファイルに含まれているクリーン情報ファイルからコマンドラインを抽出して実行することができる。

【0224】なお、圧縮後の関連付け情報としては、図26（D）の例のように、書庫ファイル番号<##>のよう

な各種の情報を含めることができる。一般に、圧縮後の関連付け情報は、拡張子を示す情報と、その拡張子に関連付けられたプログラムの実行ファイルを含む書庫ファイルを示す情報と、解凍された実行ファイルを起動するためのコマンドラインを示す情報とを含むように作成すればよい。

【0225】ステップS106では、選択されたファイルや情報が削除（アンインストール）される。なお、個々のファイルや情報を「項目」と呼ぶこともある。

【0226】図28のステップS101の処理は、この発明におけるアプリケーション選択手段の機能に相当する。また、ステップS102は検索手段に、ステップS103は書庫化手段に、ステップS104はアイコン変更手段に、また、ステップS106は削除実行手段の機能にそれぞれ相当する。

【0227】図29は、書庫ファイルの利用手順を示すフローチャートである。ステップS111では、ユーザが解凍用アイコンをダブルクリックして、書庫化されたアプリケーションの解凍を指示する。書庫化アンインストール部300は、これに応じて解凍プログラム（すなわちアークファイル部208）を起動する。ステップS112では、ユーザに解凍の確認をとるための画面が表示される。この理由は、間違って解凍用アイコンを起動した時に解凍処理を中止するためである。解凍を行なうことが確認されると、ステップS113において、アークファイル部208が書庫ファイルを解凍する。この際、解凍処理中であることを示す特別のアイコンが画面に表示されるとともに、解凍処理の進行状況を示すグラフィック表示も表示される。なお、解凍されたアプリケーションはハードディスク84内に格納される。

【0228】ステップS114では、解凍されたアプリケーションが起動され、ステップS115では、そのアプリケーションが実行される。解凍されたアプリケーションの実行中は、書庫化アンインストール部300がそのままメインメモリ25内に常駐しており、アプリケーションの終了を監視している。この理由は、ユーザが書庫化アンインストール部300の動作を終了させてしまうことを防止して、書庫化アンインストール部300が後述するステップS117～S119の処理（「後処理」と呼ぶ）を行なえるようにするためである。

【0229】ステップS116においてアプリケーションが終了したことを検出すると、ステップS117において、アプリケーションを解凍状態で格納しておくか、書庫化した状態に戻すかをユーザに問い合わせる。解凍状態で格納しておく場合には、ステップS118において書庫ファイルと解凍用アイコンを削除して、すべての処理を終了する。この時には、ハードディスク84内に格納された解凍後のアプリケーションとその関連ファイルがそのまま保存される。

【0230】一方、書庫化した状態に戻す場合には、ス

テップS119において、書庫ファイルの更新を行なう。この更新処理では、ステップS115におけるアプリケーションの実行中に変更されたファイルや情報のみを再圧縮して、再圧縮された部分が元の書庫ファイルの対応部分と書き換えられる。例えば、アプリケーションの環境設定が変更された場合には、その環境設定が登録されているINIファイルが再圧縮されて更新される。但し、アプリケーションの実行（ステップS115）において作成されたドキュメントファイルは圧縮の対象から除外するようにしてもよい。解凍されたファイルや情報の種類は、書庫ファイルに含まれているクリーン情報ファイルを参照することによって知ることができる。また、それらが変更されたか否かはそれらの最終更新日を調べることによって知ることができる。従って、クリーン情報ファイルに登録されているファイルや情報の中で、変更されたものだけを再圧縮するようにすることが可能である。書庫ファイルにはクリーン情報ファイルが含まれているので、再圧縮の際にはアプリケーションの解析（図28のステップS102）を行なう必要はない。

【0231】なお、書庫ファイルには、圧縮されているファイルや情報の位置とサイズとを示す管理データが登録されている。従って、この管理データを参照すれば、書換の対象となっているファイルや情報の圧縮データが、書庫ファイル内のどの位置に存在するかを知ることができる。

【0232】図29の処理の全体は、この発明における解凍手段の機能に相当する。また、特にステップS119の処理は、書庫ファイル更新手段の機能に相当する。

【0233】このように、書庫化アンインストール部300は、アプリケーションの解析（ステップS102）によって削除候補とされたファイルおよび情報のすべてと、クリーン情報ファイル（これは削除候補となったファイルおよび情報の削除内容を示す情報である）とを書庫化した後に削除（アンインストール）を行なう。なお、オペレーティングシステムが提供するDLLファイルは削除候補とはならないので、解析によって削除候補とされたファイルや情報をすべて削除しても、オペレーティングシステムや他のアプリケーションの動作に悪影響を及ぼすことはない。

【0234】書庫ファイルは、削除候補とされたすべてのファイルと情報を含んでいるので、この書庫ファイルを解凍することによって、そのアプリケーションを確実に起動することができる。また、書庫ファイルを作成した際に、その書庫ファイルを解凍するための解凍用アイコンを登録するようにしたので、ユーザはそのアイコンを利用して、書庫化されているアプリケーションを容易に起動することができる。

【0235】L. アプリケーションの転送への応用：上記の書庫ファイルは、コンピュータ間のアプリケーション

ンの転送に利用することが可能である。図30は、書庫ファイルを利用したコンピュータ間のアプリケーションの転送の手順を示すフローチャートである。

【0236】転送元コンピュータでは、転送するアプリケーションの選択(S121)と、解析(S122)と、削除候補(書庫化候補)の圧縮(S123)とを行なう。これらのステップS121~S123は、前述した図28のステップS101~S103と同じである。但し、図30の手順では、ファイルや情報のアンインストールは行なわないので、「削除候補」という呼び方は正確ではなく、むしろ「書庫化候補」と呼ぶべきである。

【0237】転送先コンピュータでは、ファイル整理プログラム群(図2参照)をインストールし(ステップS131)、書庫化アンインストール部300(図27)を起動する(S132)。そして、フロッピディスクやオンラインで転送された書庫ファイルを指定して解冻を行なう(S133)。

【0238】書庫ファイルの解冻においては、ドライブとディレクトリの取扱いに注意を要する。すなわち、転送元コンピュータに存在したドライブやディレクトリが、転送先コンピュータには存在しない場合がある。このような場合には、どのドライブのどのディレクトリに解冻ファイルを格納すべきかをユーザに問い合わせ、ユーザの指定に応じて格納するようにすればよい。

【0239】なお、図30のステップS121の処理は、この発明におけるアプリケーション選択手段の機能に相当する。また、ステップS122は検索手段に、ステップS123は書庫化手段に、ステップS133はアイコン変更手段の機能にそれぞれ相当する。

【0240】書庫ファイルはデータ量が極めて小さいので、上述の手順によって簡単にアプリケーションを転送することができる。また、書庫ファイルには、転送元コンピュータにおけるアプリケーションの設定も含まれているので、転送先コンピュータにおいて転送元コンピュータと同じ設定状態を復元することができるという利点がある。

【0241】なお、この発明は上記実施例に限られるものではなく、その要旨を逸脱しない範囲において種々の態様において実施することが可能であり、例えば次のような変形も可能である。

【0242】(1)ユーザがファインダ部202において実行ファイルを削除対象として選択し、その削除を実行する場合には、ファインダ部202からクリーナ部204を呼び出して図10に示すアンインストール処理を実行するようにすることも可能である。例えば、図5に示すファイルの一覧表示から実行ファイルを削除する旨をユーザが入力すると、クリーナ部204が起動して、プログラムマネージャ内に登録されているすべてのアイコンの登録内容を図13のステップS11と同様な手順

で解析する。そして、ファインダ部202で選択された実行ファイルが削除対象メインファイルとなるようなアイコンを決定する。その後、そのアイコンと削除対象メインファイルとに関して図10のステップS3~S9が実行される。

【0243】このように、ファインダ部202からクリーナ部204が呼び出されるようにすれば、ファインダ部202を用いて不要と判断された実行ファイルの削除に伴って、同時に不要となる種々のファイルや情報を容易に削除することができ、ハードディスクの内容をより効率的に整理することができる。

【0244】(2)ファイルの削除に使用されるアクセス履歴はすべて履歴記録部212で作成される必要はなく、例えば、オペレーティングシステムで管理されている最終アクセス日時等を利用することも可能である。この場合には、履歴記録部212は、オペレーティングシステムで管理されているアクセス履歴以外の履歴を登録するようにすればよい。

【0245】(3)クリーナ部204を利用してファイルを削除する場合に、アクセス履歴に基づいて削除候補とするファイルを検索・抽出するようにすることも可能である。また、図10のステップS4において、アクセス履歴に基づいてデフォルト削除候補を決定するようにすることも可能である。例えば、アプリケーションのディレクトリ内のファイルに関してデフォルト削除候補を決定する場合に、アプリケーションをインストールした日時以降にアクセスされていないファイルをアクセス履歴から検索して、そのファイルを削除候補とすることができる。

#### 【図面の簡単な説明】

【図1】この発明の一実施例を適用するコンピュータシステムのハードウェア構成を示すブロック図。

【図2】ハードディスク84内のファイルの整理を行なうプログラム群の機能を示す機能ブロック図。

【図3】ディレクトリ別一覧の画面表示の例を示す説明図。

【図4】拡張子別一覧の画面表示の例を示す説明図。

【図5】アクセス履歴を含むファイル一覧の画面表示の例を示す説明図。

【図6】アクセス履歴における利用度の分類例を示す説明図。

【図7】ファイルの詳細情報の画面表示の例を示す説明図。

【図8】WIN.INIのファイル内容の一例を示す説明図。

【図9】図9は、クリーナ部204の内部構成を示すブロック図である。

【図10】アンインストール処理の手順を示すフローチャート。

【図11】アプリケーションの選択画面(選択ダイアログ)を示す説明図。

【図12】実施例におけるディレクトリ構造を示す説明図。

【図13】ステップS3の詳細手順を示すフローチャート。

【図14】ステップS11の詳細手順を示すフローチャート。

【図15】コマンドラインの例を示す説明図。

【図16】ステップS13の詳細手順を示すフローチャート。

【図17】インポート名テーブルとリロケーションテーブルの構成を示す説明図。

【図18】DLL ファイルとVBX ファイルの参照リストの例を示す説明図。

【図19】登録データベース内のOLE 情報の登録内容の例を示す説明図。

【図20】削除候補の内容を示す説明図。

【図21】参照リンク240の登録内容を示す説明図。

【図22】削除候補の解析結果を示す画面の一例を示す説明図。

【図23】削除対象の確認ウィンドウの一例を示す説明図。

【図24】クリーン情報ファイルの内容を示す説明図。

【図25】参照リンク240の作成手順を示すフローチャート。

【図26】レジストリファイルに登録されている関連付け情報を示す説明図。

【図27】書庫化アンインストール部300の機能を示す説明図。

【図28】書庫化アンインストール処理の手順を示すフローチャート。

【図29】書庫ファイルの利用手順を示すフローチャート。

【図30】書庫ファイルを利用したコンピュータ間のアプリケーションの転送の手順を示すフローチャート。

【符号の説明】

10…コンピュータ

20…演算処理部

21…CPU

22…ローカルバス

23…キャッシュメモリ

24…キャッシュコントローラ

25…メインメモリ

30…PCIブリッジ

32…PCIバス

40…コントローラ部

42…ISAバス

44…VGA

46…SCSIコントローラ

48…ISAブリッジ

54…複合I/Oポート

56…サウンドI/O

60…I/O部

62…ISAスロット

68…タイマ

72…キーボード

73…マウス

74…スピーカ

76…CRT

82…フロッピディスク装置

84…ハードディスク

86…パラレルポート

88…プリンタ

90…シリアルポート

92…モデム

96…マイクロフォン

202…ファインダ部

204…クリーナ部

206…リソース監視部

208…アークファイラ部

210…レスキュー部

212…履歴記録部

220…アクセス監視部

222…履歴ログ

224…履歴データベース更新部

226…履歴データベース

230…ユーザインタフェイス部

232…情報取得部

234…解析部

236…アンインストール部

238…バックアップ部

240…参照リンク

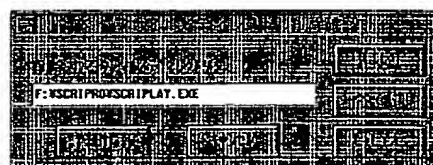
300…書庫化アンインストール部

【図19】

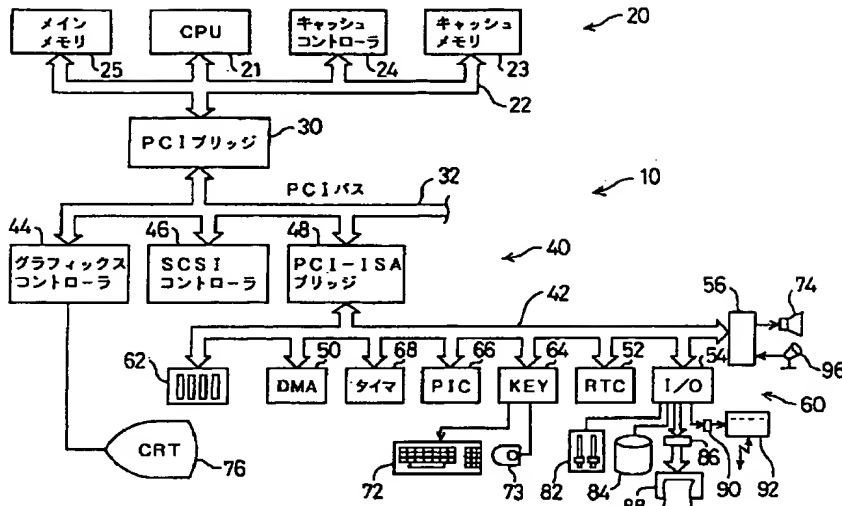
登録データベース「REG.DAT」内のOLE情報

```
.dxt = Disk2.Document
Disk2.Document = Disk2 |'ファイル'
Disk2.Document\protocol\StdFileEditing\server = A:\DISK2\XUTW.EXE
```

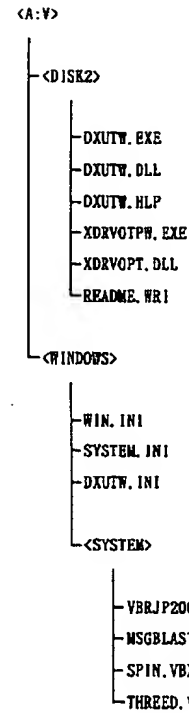
【図23】



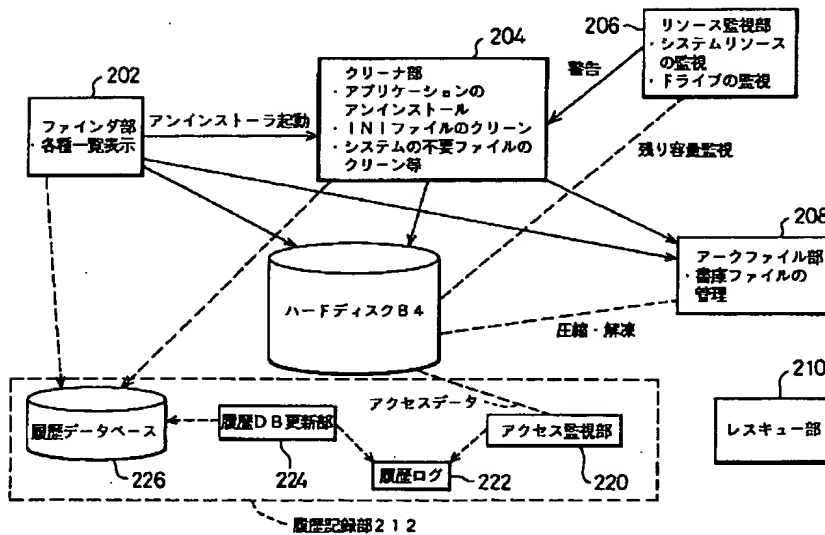
【図1】



【図12】



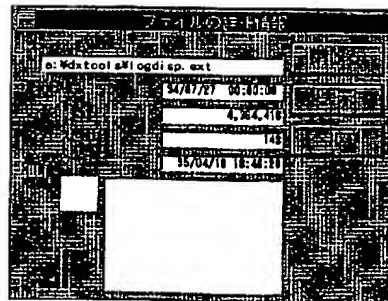
【図2】



【図6】

【図7】

利用度	
○ よく利用する	10日以内にアクセスされ、回数が1回以上のファイル 30日以内にアクセスされ、回数が5回以上のファイル 90日以内にアクセスされ、回数が25回以上のファイル
○ 割と利用する	30日以内にアクセスされ、回数が1回以上のファイル 90日以内にアクセスされ、回数が5回以上のファイル
△ 利用することもある	90日以内にアクセスされ、回数が1回以上のファイル
なし 利用していない	アクセス回数が0回のファイル 90日以上アクセスされていないファイル



【図3】

●ディレクトリ別一覧

ファイル名	属性	バイト	作成日時	更新日時
AAA		2	112,687	0.83
ADDITION		77	2,374,118	0.83
ADDITION		80	2,374,118	0.83
ADDITION		11	2,374,118	0.83
ADDITION		171	2,374,118	1.03
ADDITION		83	2,374,118	0.83
ADDITION		88	2,374,118	0.83
ADDITION		15	2,374,118	1.03
ADDITION		8	2,374,118	0.83
ADDITION		17	2,374,118	0.83
ADDITION		17	2,374,118	0.83

【図5】

ファイナダー

ファイル名	サイズ	作成日時	更新日時
112687001.exe	112,687	01:45:18	112687001.exe
112687002.exe	112,687	01:45:18	112687002.exe
112687003.exe	112,687	01:45:18	112687003.exe
112687004.exe	112,687	01:45:18	112687004.exe
112687005.exe	112,687	01:45:18	112687005.exe
112687006.exe	112,687	01:45:18	112687006.exe
112687007.exe	112,687	01:45:18	112687007.exe
112687008.exe	112,687	01:45:18	112687008.exe
112687009.exe	112,687	01:45:18	112687009.exe
112687010.exe	112,687	01:45:18	112687010.exe

【図8】

## WIN.INIの内容

【図4】

●拡張子別一覧

ファイル名	属性	バイト	作成日時	更新日時
AAA		2	112,687	0.83
ADDITION		77	2,374,118	0.83
ADDITION		80	2,374,118	0.83
ADDITION		11	2,374,118	0.83
ADDITION		171	2,374,118	1.03
ADDITION		83	2,374,118	0.83
ADDITION		88	2,374,118	0.83
ADDITION		15	2,374,118	1.03
ADDITION		8	2,374,118	0.83
ADDITION		17	2,374,118	0.83
ADDITION		17	2,374,118	0.83

[windows] ← セクション名

spooler=yes

load= 値

run= キー

Beep=yes

...

[Desktop]

Pattern=

Wallpaper=

GridGranularity=0

...

[Extensions]

cal=calendar.exe \*.cal

txt=notepad.exe \*.txt

hlp=winhelp.exe \*.hlp

dxt=a:YDISKZYDZUTW.EXE \*.dxt

...

[DXUTW]

DSKJET=+1

Backup=0

...

ウィンドウ全般の設定

背景パターン、壁紙等の設定

拡張子とアプリケーションの  
実行ファイルの関連付け情報

アプリケーションの設定

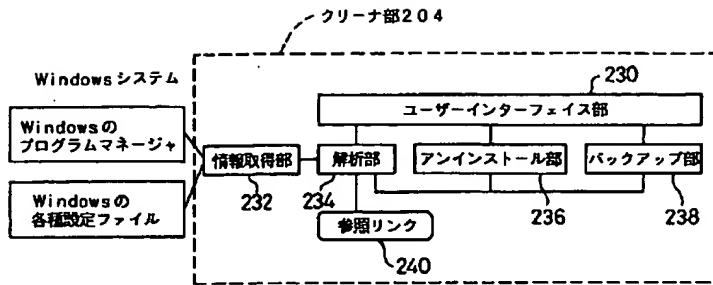
【図11】

アプリケーションウィンドウ

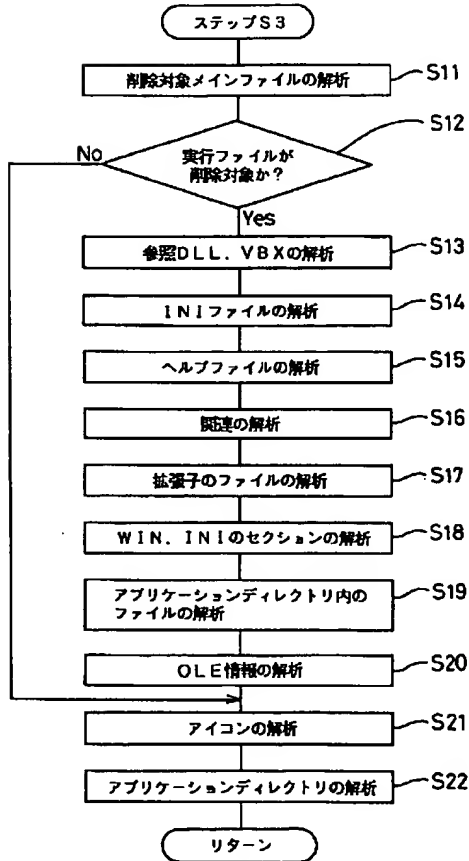
ファイル名	属性	バイト	作成日時	更新日時
AAA		2	112,687	0.83
ADDITION		77	2,374,118	0.83
ADDITION		80	2,374,118	0.83
ADDITION		11	2,374,118	0.83
ADDITION		171	2,374,118	1.03
ADDITION		83	2,374,118	0.83
ADDITION		88	2,374,118	0.83
ADDITION		15	2,374,118	1.03
ADDITION		8	2,374,118	0.83
ADDITION		17	2,374,118	0.83
ADDITION		17	2,374,118	0.83



【図9】

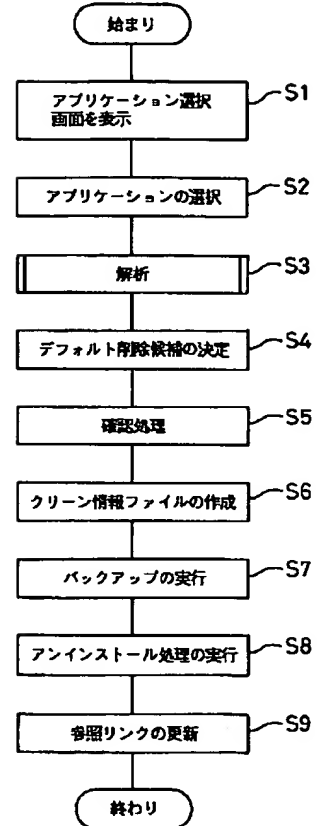


【図13】



【図10】

アンインストール処理フロー



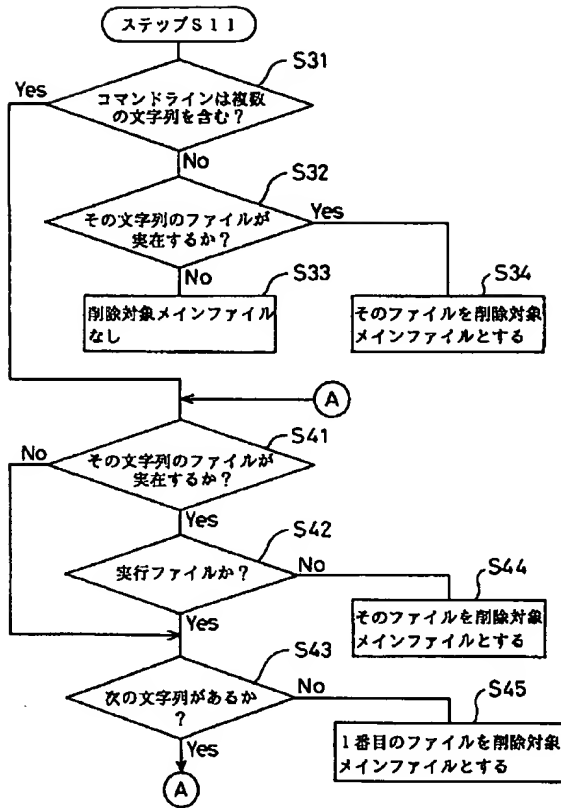
【図18】

A: WD1SK2V0XUTW.EXEが削除対象メインファイルの場合の参照リスト

```

A: %WINDOW%\SYSTEM\WVBRJP200.DLL
A: %WINDOW%\SYSTEM\WVSGBLAST.VBX
A: %WINDOW%\SYSTEM\WVSPIN.VBX
A: %WINDOW%\SYSTEM\WVTHREED.VBX
A: %WD1SK2V0XUTW.DLL
  
```

【図14】



【図15】

(A) 例1. マントライン

WINHELP.EXE A:VDISK2VXDUTW.HLP

↑ 存在

削除対象メインファイル

・ A:VDISK2VXDUTW.HLPが存在 → A:VDISK2VXDUTW.HLP

・ A:VDISK2VXDUTW.HLPが存在せず → A:WINDOWS\WINHELP.EXE

(Windowsのパスが順で検索してデフォルトパスを得る)

(B) 例2. マントライン

A:VTEST.EXE /P /U C:VTTT.DAT D:YUUU.TXT

↑ 存在

削除対象メインファイル

・ C:VTTT.DATが存在 → C:VTTT.DAT

・ C:VTTT.DATが存在せず、D:YUUU.TXTが存在 → D:YUUU.TXT

・ C:VTTT.DATが存在せず、D:YUUU.TXTも存在しない → A:VTEST.EXE

(A:VTEST.EXEが存在しなければ、削除対象メインファイルなし)

(C) 例3. マントライン

Y:VCCMAIL\JCCMAIL\VMMAIL.EXE Y:VCCMAIL\VCCDATA "MICHEAL" PASSWORD

↑ 存在

↑ 存在せず

削除対象メインファイル

Y:VCCMAIL\JCCMAIL\VMMAIL.EXE

【図21】

参照リンク240の登録内容

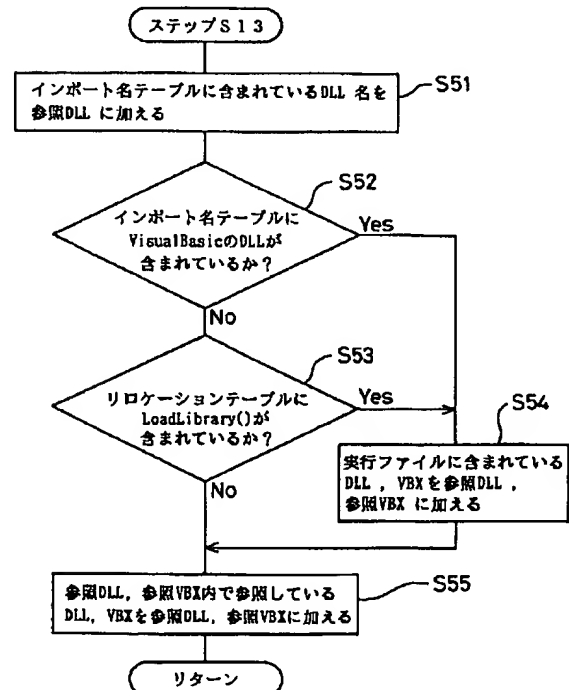
(A)

実行ファイルリスト
A:VDISK2VXDUTW.EXE
A:VDISK2VDRVOTPW.EXE
...

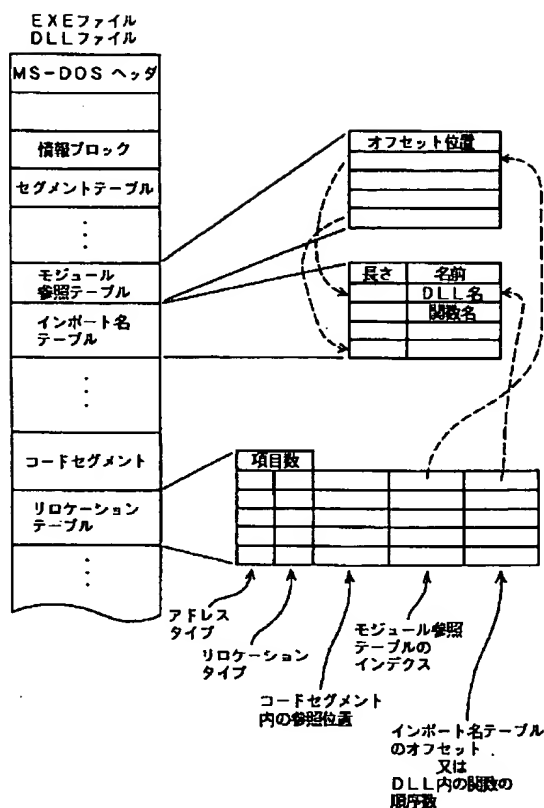
(B)

被参照ファイルリスト	参照数
A:WINDOWS\SYSTEM\VBVBRJP200.DLL	10
A:WINDOWS\SYSTEM\MSGBLAST.VBX	8
A:WINDOWS\SYSTEM\WSPIN.VBX	6
A:WINDOWS\SYSTEM\WTHREED.VBX	8
A:VDISK2VXDUTW.DLL	2
A:VDISK2VDRVOTPW.DLL	1
...	

【図16】



【図17】



【図20】

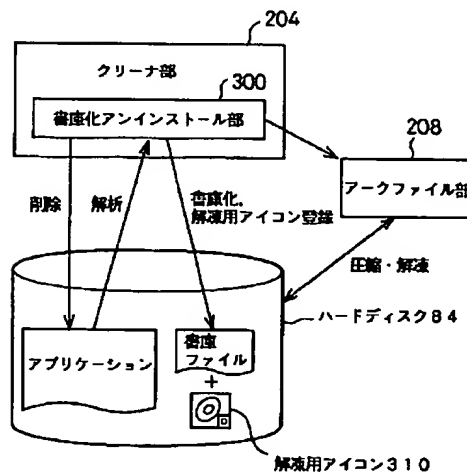
- 削除対象の内容
- (A) 削除対象メインファイル  
A:VDISK2\DXUTW.EXE
- (B) 参照DLL, 参照VBX  
A:WINDOWS\SYSTEM\VB2\JP200.DLL  
A:WINDOWS\SYSTEM\MSGLAST.VBX  
A:WINDOWS\SYSTEM\SPIN.VEX  
A:WINDOWS\SYSTEM\THREED.VBX  
A:VDISK2\DXUTW.DLL
- (C) INIファイル  
A:WINDOWS\DXUTW.INI
- (D) ヘルプファイル  
A:VDISK2\DXUTW.HELP
- (E) WIN.INI内の関連付け情報  
dxt=a:VDISK2\DXUTW.EXE ".dxt
- (F) アプリケーションに関連付けられた拡張子を有するファイル  
なし
- (G) WIN.INI内のアプリケーションセクション  
[DXUTW]
- (H) アプリケーションディレクトリ内の他のファイル  
「ディレクトリ内のファイル A:VDISK2 (個別に選択する)」
- (I) 登録データベース内のOLE情報  
.dxt = Disk2.Document  
Disk2.Document = Disk2 ドキュメント  
Disk2.Document\protocol\StdFileEditing\server = A:VDISK2\DXUTW.EXE
- (J) プログラムマネージャ内のアイコンの登録内容  
「DISK2 ユーティリティ」の登録内容
- (K) アプリケーションディレクトリ  
「ディレクトリ A:VDISK2 (ディレクトリが空になったら削除)」

【図24】

#### クリーン情報ファイルの内容

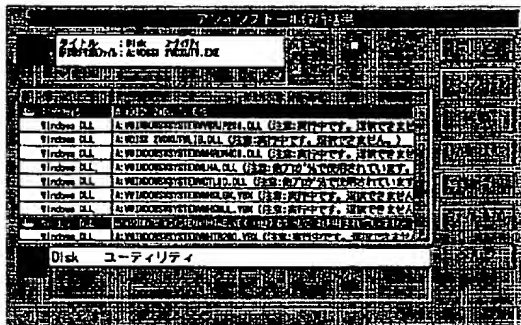
- (A) INIファイルのセクションの削除 (図19 (G))  
WIN.INI内のセクション [DXUTW]
- (B) INIファイル内のキー削除 (図19 (E))  
WIN.INI内のキー「dxt=a:VDISK2\DXUTW.EXE ".dxt」
- (C) 登録データベース内の情報の削除 (図19 (I))  
.dxt = Disk2.Document  
Disk2.Document = Disk2 ドキュメント  
Disk2.Document\protocol\StdFileEditing\server = A:VDISK2\DXUTW.EXE
- (D) プログラムマネージャ内のアイコン削除 (図19 (J))  
アイコン「DISK2 ユーティリティ」の登録内容  
・グループ名  
・タイトル  
・コマンドライン  
・実行時のディレクトリ  
・アイコンファイル
- (E) プログラムマネージャのグループ削除 (グループを削除した場合)  
削除したグループ名

【図27】

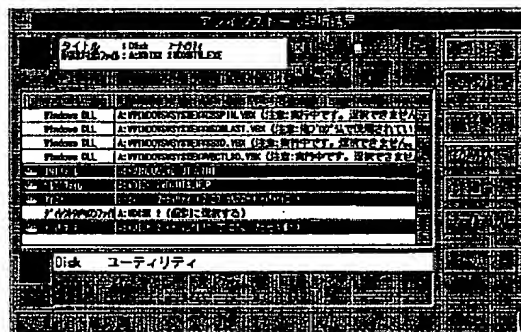


【図22】

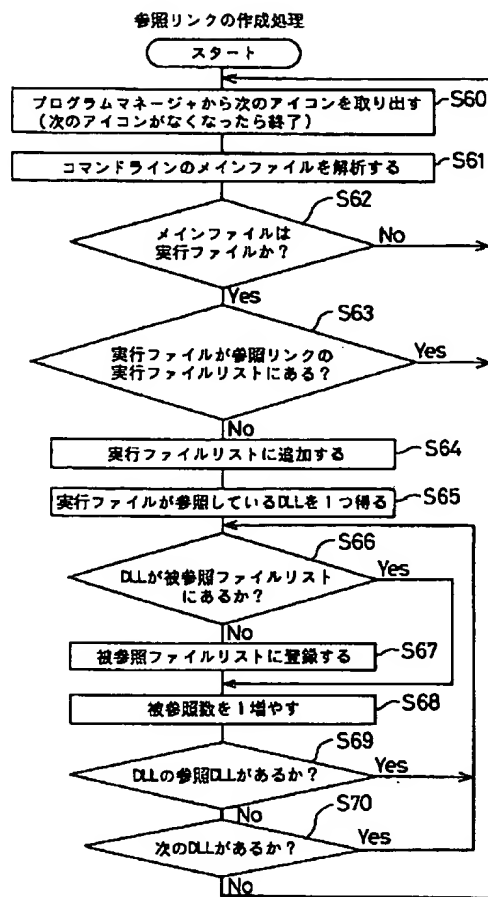
(A)



(B)



【図25】



【図26】

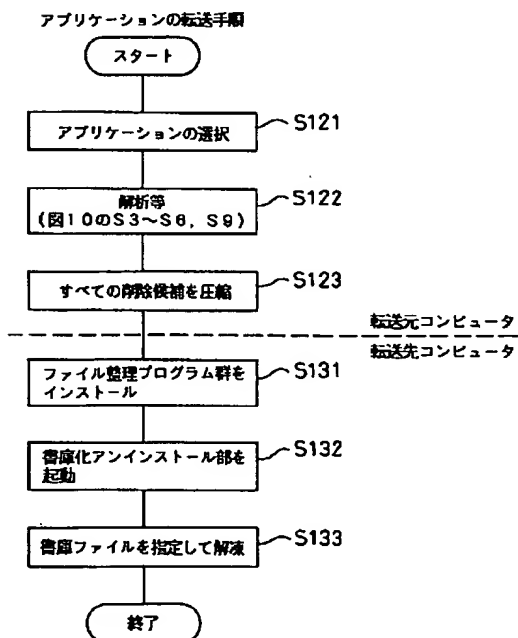
(A) レジストリファイル内の関連付け情報のデータ形式  
 <拡張子> = <ドキュメントタイプ>  
 <ドキュメントタイプ> = <ドキュメント名>  
 <ドキュメントタイプ> %shell\open\command = <実行ファイル> "%1"

(B) 関連付け情報の例  
 .dx2 = Disk2.Document  
 Disk2.Document = Disk2 ドキュメント  
 Disk2.Document %shell\open\command = a:\DISK2\VDXUTW.EXE "%1"

(C) 圧縮後の関連付け情報  
 .dx2 = Disk2.Document  
 Disk2.Document = Disk2 ドキュメント  
 Disk2.Document %shell\open\command = C:\VDXT\VDXMELT.EXE /MELT  
 "ARK.ZIP" /ARC "a:\DISK2\VDXUTW.EXE %1"

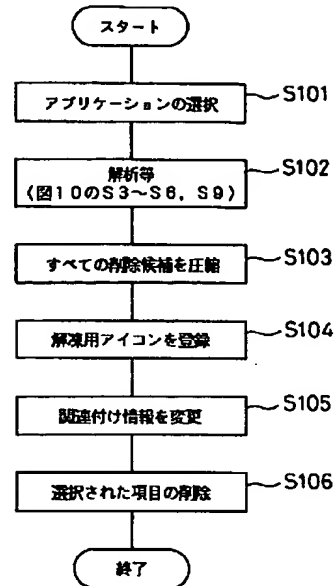
(D) レジストリファイル内の関連付け情報の他の例  
 .dx2 = Disk2.Document  
 Disk2.Document = Disk2 ドキュメント  
 Disk2.Document %shell\open\command = C:\VDXT\VDXMELT.EXE <##>  
 /ARC

【図30】



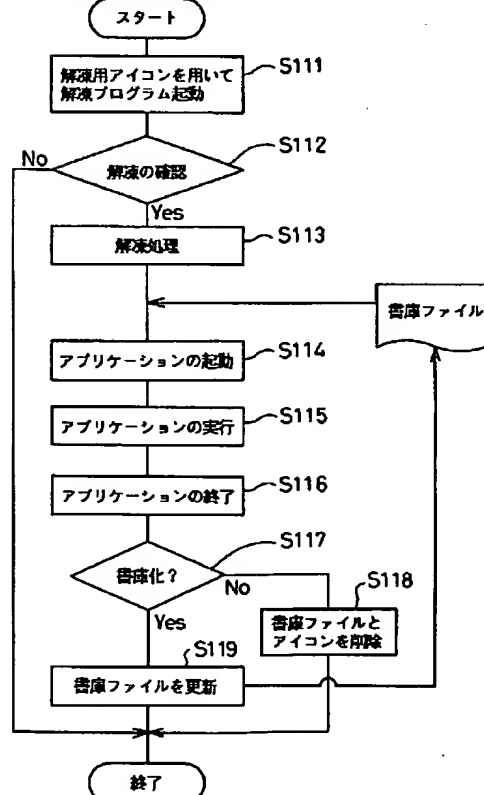
【図28】

書庫化アンインストール処理の手順



【図29】

書庫ファイルの利用手順



フロントページの続き

(72)発明者 牛山 憲一  
長野県松本市中央二丁目1番27号 エー・  
アイ ソフト株式会社内

(72)発明者 富樫 宏規  
長野県松本市中央二丁目1番27号 エー・  
アイ ソフト株式会社内